Design • Build • Program • Learn • Compete

SERVO MAGAZINE

FOR THE ROBOT INNOVATOR

www.servomagazine.com

November 2011

Do you speak LINGODROID?

These robots chat it up with a language all their own

- VEX VR Control Glove
- Man vs. Machine
- Digilent USB I/O Explorer

U.S. $5.50 CANADA $7.00

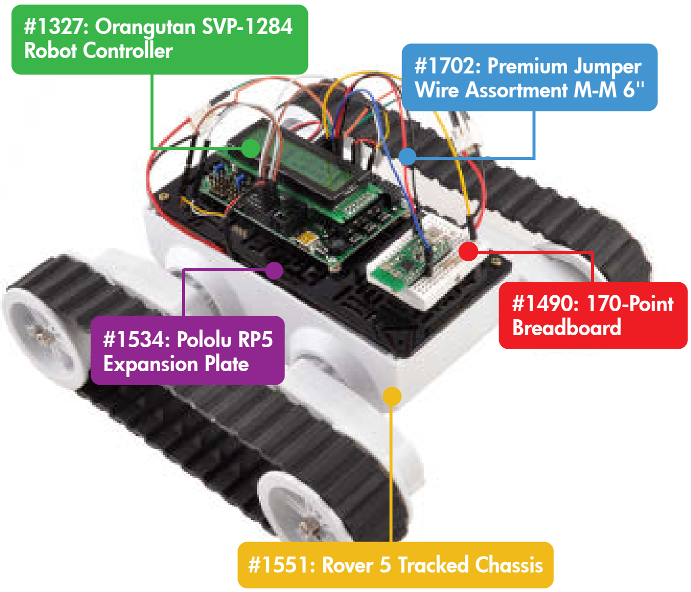
6.0 Volts		7.4 Volts	
Speed	Torque	Speed	Torque
0.16	278 oz-in	0.13	347 oz-in

Part#	Dimensions	Weight
35765S	2.32 x 1.14 x 2.04 in	6.07 oz

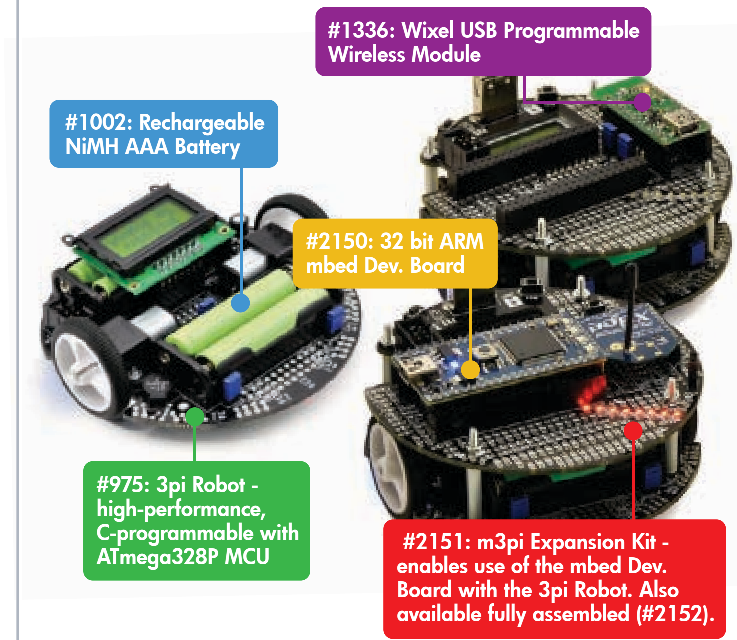
HITEC

12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecrcd.com

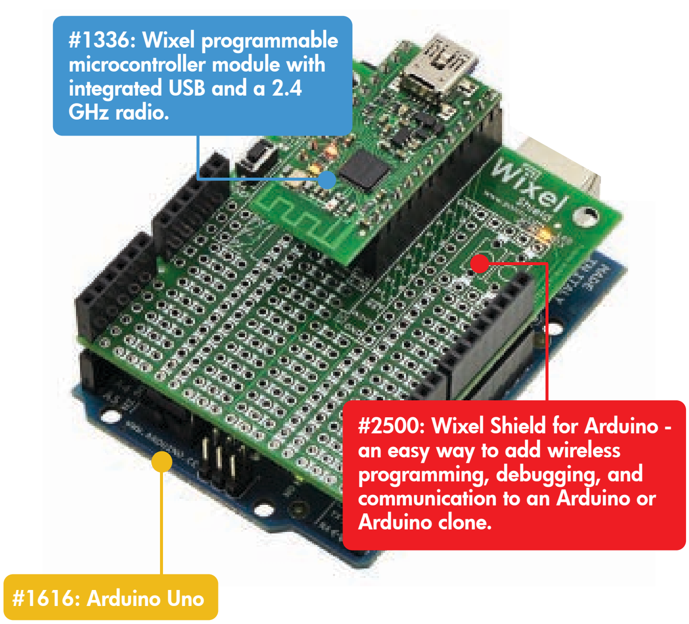
DIY Projects:
Simple Tracked Robot



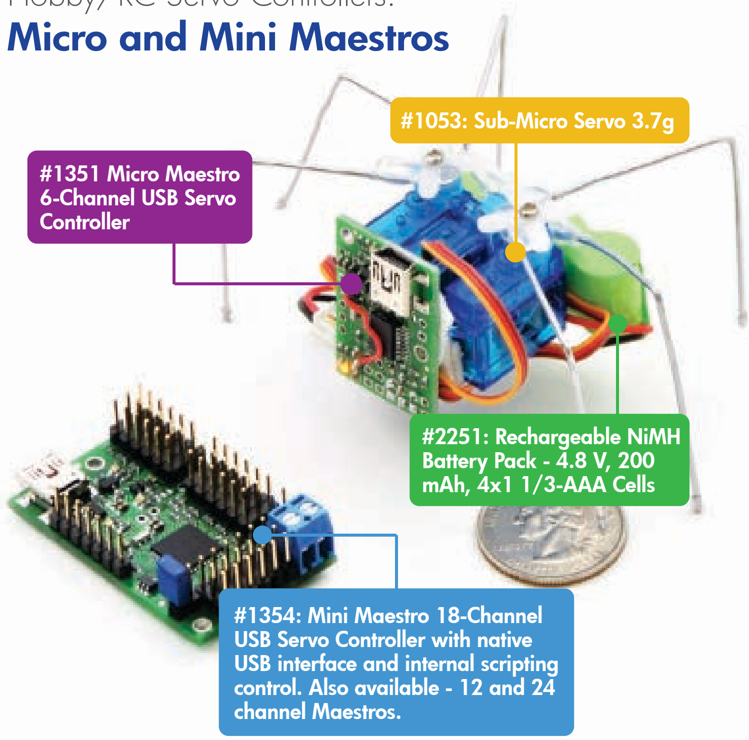
Robots and Robot Kits:
Pololu 3pi and m3pi



Programmable Controllers:
Wixel and Wixel Shield



Hobby/RC Servo Controllers:
Micro and Mini Maestros

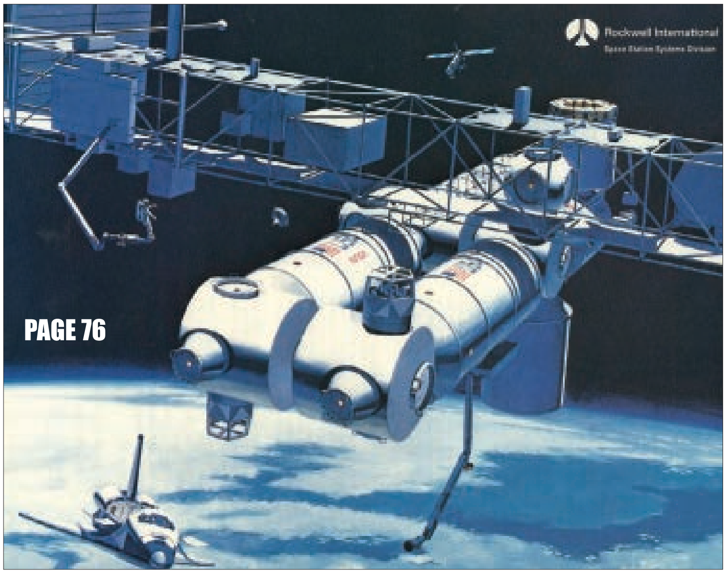


Finding the right parts for your robot can be difficult, but you also don't want to spend all your time reinventing the wheel (or motor controller). That's where we come in: Pololu has the unique products - from actuators to wireless modules - that can help you take your robot from idea to reality.

SERVO MAGAZINE

11.2011

VOL. 9 NO. 11



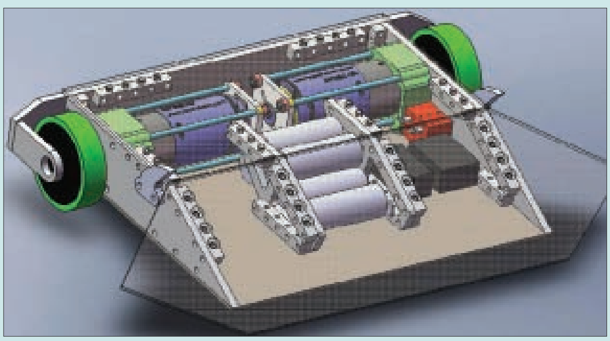
Interact With Writers and Other Readers at the SERVO Forums!

Now you can discuss all your favorite articles in the magazine at our newly redesigned forums. Just point your browser to **http://forum.servomagazine.com** where you can post comments, engage with other readers, ask questions, and talk to the writers.

Columns

The Combat Zone...

Features

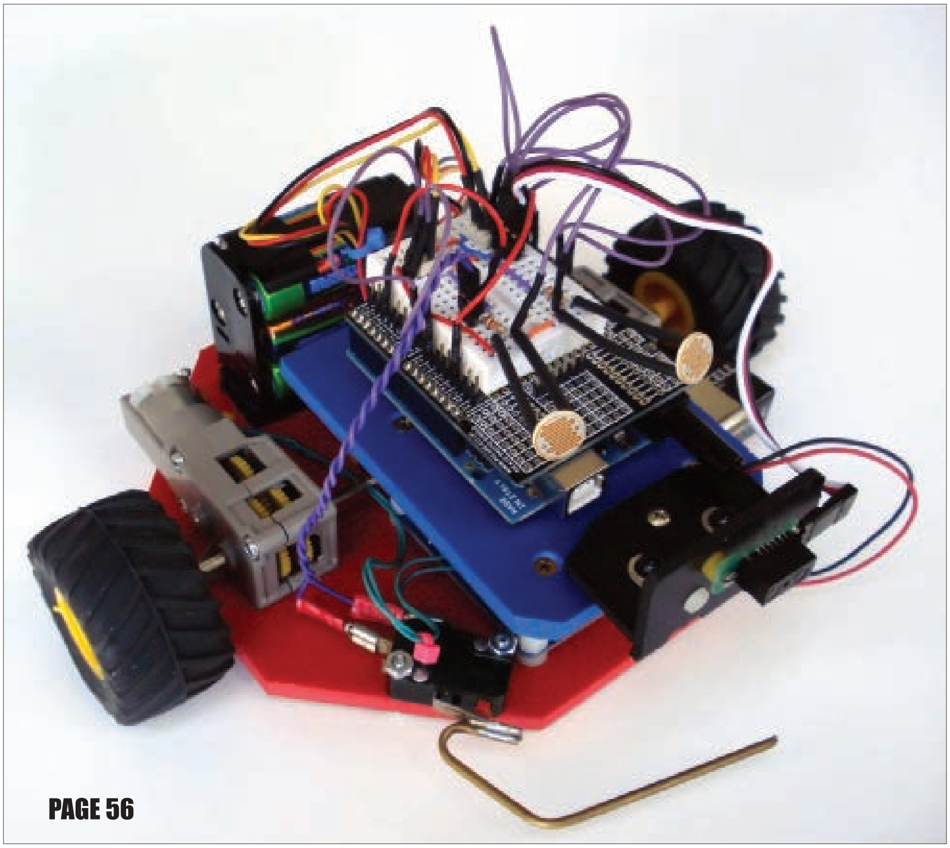


Events

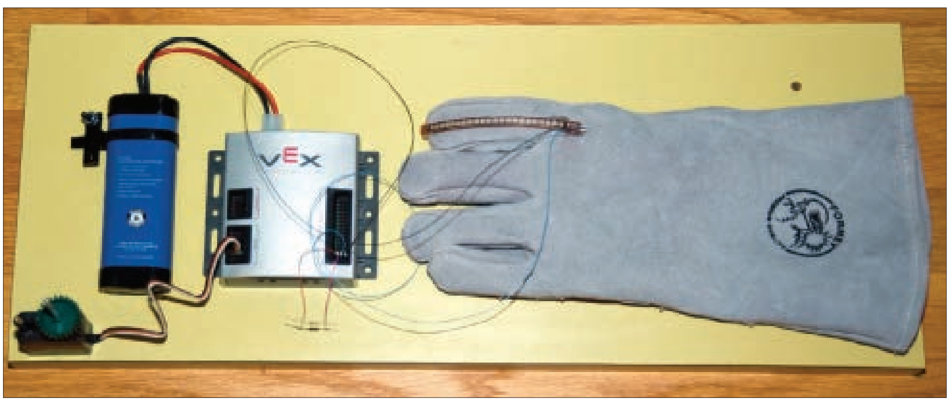
Departments

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

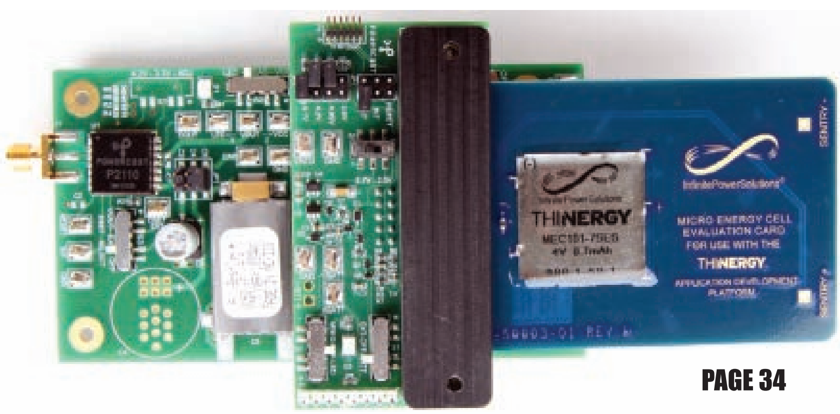
In This Issue ...



PAGE 56



PAGE 48



PAGE 34

Mind / Iron

by Bryan Bergeron, Editor

Re-inventing the Wheel

The field of robotics is so vast – spanning mechanical, electrical, and computer engineering – that you have to focus on one area to get anything done. For example, if you're into pattern recognition-based navigation, you'll have your hands full with image processing algorithms and perhaps image acquisition hardware design.

That said, it pays to occasionally broaden your focus and take a look at what's happening outside the world of robotics. There's a lot of activity in the electronics industry that can be applied directly to robotics, saving you time and money.

For example, I was looking for a way to quickly illuminate a scene with IR, alternating with visible light in order to improve a vision-based navigation system. I started with two banks of LEDs and an Arduino Mega processor. However, a few weeks later — while searching for LEDs on eBay — I happened across several inexpensive LED light arrays, all compatible with DMX512 controllers. Up to that point, I hadn't heard of DXM.

It turns out that the DXM (Digital Multiplex) standard has been around for decades, and that it's used to control dimmers, fog machines, and moving lights. You can check out the standard on the Web, but the point is that I was just about to replicate a small part of the DXM standard, unaware that I could buy — off the shelf — exactly what I needed. Compact controllers with an assortment of sliding and rotating potentiometers and buttons are readily available.

More importantly, for my project I located an inexpensive Arduino-compatible DXM board. The microcontroller board enables me to control the IR and visible LED banks through a USB link with my laptop, as well as run the banks autonomously.

Are you in the process of unknowingly re-inventing a protocol or algorithm that's been around for years? There's something to be said for what can be learned by tackling a tough software or hardware problem. The problem is that time is limited, and you're probably better off buying infrastructure devices and focusing on your strengths and interests.

Also, when you're trying to solve a problem, the solution isn't always selling on eBay or Amazon for a few dollars. It may be the problem and resulting solution were addressed in the past, with little or no commercial footprint. Take the hemispherical omnidirectional gimbaled wheel, or HOG wheel. It's been around for nearly a century, but unless you follow RoboGames you probably don't know how it works, or how you can apply it to robotics.

According to IEEE Spectrum (**spectrum.iee.org/blog/automation**) Curtis Boirum, a graduate student at Bradley University demonstrated a drive system based on the HOG wheel at the 2011 RoboGames symposium. The drive is a rubber hemisphere that rotates about the vertical axis, with servos that can tilt the axis left and right and forward and backwards. You can see two versions of the robot in action by searching for "Hemispherical Gimbaled Wheel Drive System" on

FOR THE ROBOT INNOVATOR
SERVO MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Jenn Eckert, Tom Carroll, David Geer, Dennis Clark, R. Steven Rainwater, Kevin Berry, Gordon McComb, Fred Eady, Joe Grand, Daniel Ramirez, Mike Jeffries, Pete Smith, Bryce Woolley, Evan Woolley

CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

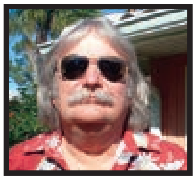
WEB CONTENT
Michael Kaudze
website@servomagazine.com

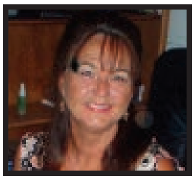
ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Shannon Christensen

YouTube. The speed and agility of the robot are amazing — there's no way a human could evade this robot.

So, how can you keep track of both the past and the increasingly rapid rate of current innovation? I set aside one day a month to catch up on the old and the new. I'm fortunate to have several engineering libraries in my area, with books on past inventions. When I can't make it to a library, I search the patent database at **USPTO.gov**. For current innovations, it's hard to beat Google and — of course — contemporary magazines such as *SERVO*. **SV**

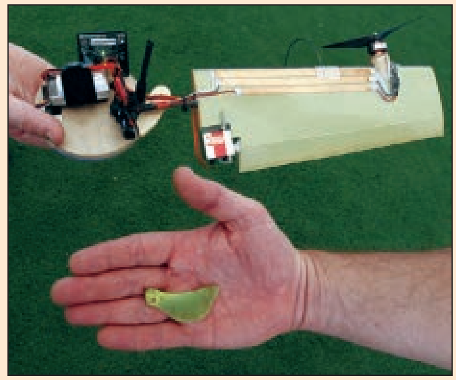




Robytes

by Jeff and Jenn Eckert

Whirlybot Imitates Maple Seed



New Lockheed Martin UAV spins up (and down).

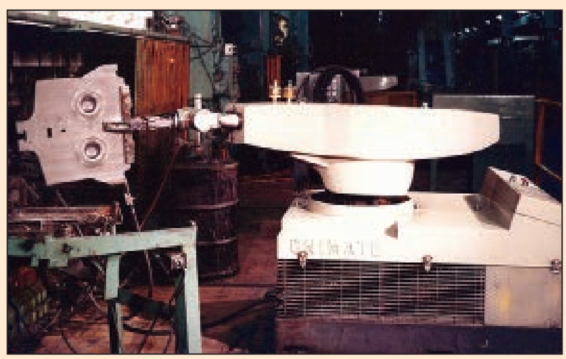
Most of us remember sitting under a maple tree and watching the little whirlybird seeds — technically called "samaras" — spinning around as they drifted to earth. The associated aerodynamic principles have now been applied by Lockheed Martin's Advanced Technology Laboratories (**www.atl.lmco.com**) to a new UAV. Demonstrated at the Association for Unmanned Vehicle Systems International Conference this summer, the "Samarai" is a vertical takeoff and landing device that provides stable hovering and on-board video streaming — all from a package that weighs less than half a pound and has only two moving parts. According to the company, the design is scalable, allowing it to be used for a variety of missions, including surveillance and reconnaissance and payload delivery. Because it is produced via 3-D printing, there are no expensive production costs. The Samarai can be controlled using a dedicated remote or even with an app on a tablet computer. Lockheed didn't mention it, but the 16 in (41 cm) flyer looks suspiciously similar to one developed at the University of Maryland's Autonomous Vehicle Laboratory (**www.avl.umd.edu**) and called "Samara." Presumably, they've worked something out with regard to the university's pending patents on the design. You can see the vid at **www.youtube.com/watch?v=T8FBDFJ1cbk**.

UAV Perches On Walls

UAV creeps up walls like an insect.

Another nifty airborne concept has come out of the Biomimetics and Dexterous Manipulation Lab (**bdml.stanford.edu**) at Stanford University. One problem with most flying devices used for surveillance is that they use up a lot of power while hovering. But picture a little bird that can fly up to a building, raise its nose in an intentional stall, land vertically on a wall, and hang out there for hours or days in silence, consuming practically no power. Such is the lab's "perching AUV" which eventually could be launched in intercoordinated flocks for a range of observational duties. The key is its feet which are equipped with tiny spines that "engage small asperities on the surface." In this way — using the propeller as required — the flyer can creep along the wall to achieve the best orientation for the desired view. When the mission is completed, it can jump into the air and fly home — ready for the next assignment.

R.I.P. George C. Devol



Working version of Devol's Unimate, the first industrial robot.

Sadly, we note that George C. Devol — known as the "father of robotics" — passed away on August 11 at his home in Wilton, CT, at age 99. George held the patent on the first digitally operated programmable robotic arm which became known as the Unimate. The first company to put one to work was General Motors; it was used in an assembly line to remove hot die-cast metal parts from their molds, and to lift and stack them. To see one in action, check out a 1966 episode of the *Tonight Show* at **www.youtube.com/watch?v=yKo6KMkuVAk**. The machine sinks a putt, opens and pours a can of beer, and (sort of) leads the orchestra. Other Devol inventions include the first photoelectric door opener, an early barcode system, early radar and microwave devices, robotic sensors, magnetostrictive manipulators, and even the "Speedy Weeny" — a machine for automatically cooking and vending hot dogs. In May, Devol was inducted into the National Inventors Hall of Fame (**www.invent.org**).

www.servomagazine.com/index.php?/magazine/article/november2011_Robytes

Lost Again

B9 robots and their creators, waiting for your purchase.

For reasons that not all of us comprehend, 46 years after it debuted on CBS, "Lost in Space" remains a nostalgic favorite for a multitude of fans. As one of its less enthusiastic followers noted, "Yes, it is campy. Yes, the acting is bad and the props are cheesy. Yes, it is brain-dead entertainment, but *Lost in Space* has the elements that make it very entertaining." Reportedly, most of the cast members were happy to see it go after three seasons. In fact, Guy Williams (John Robinson) was so appalled that he moved to Argentina after the show was cancelled and never acted again. However, if hearing the magic words, "Danger, Will Robinson, Danger!" still gives you a tingle, rejoice! You now can own a full size, limited edition, fully licensed reproduction of the famed Class M-3 Model B9 General Utility Non-Theorizing Environmental Control Robot. Its list of functions is too extensive to cover here, but be advised that it comes with 500 voice tracks recorded by Richard Tufeld (who provided the voice of the original unit), and his canned phrases are projected by a booming 240W stereo system. For all the details, visit **www.lostinspacerobot.com**. By the way, owning your own B9 will set you back $24,500 (including domestic shipping), so keep your credit card handy.

Look, Ma, No Farmer

If you're driving past some farms in the corn belt and think you see a John Deere slogging through the fields with no operator, your eyes may not be deceiving you. Kinze Manufacturing (**www.kinze.com**) — via the Kinze Autonomy Project — is now offering "the first truly autonomous row crop solution on this scale in the world," according to VP and chief marketer Susanne Kinzenbaw Veatch. "Knowing how important it is to get crops into the ground during the short planting window, we're excited to offer this system to help growers be productive and make the most of their harvest."

Kinze has largely eliminated the need for skilled operators in the tractor cab by replacing them with a GPS-based control system. The grower loads a field map into the system, including field boundaries and any existing obstacles such as waterways. He then drives the tractor to the field, and the system automatically figures out the most efficient game plan for planting. It positions itself at the starting point and works until the job is done — assuming it doesn't encounter any unexpected obstacles. If it does, it stops until the farmer intervenes and eliminates the problem.

In addition to planting, the equipment can be used to fertilize, maintain, and harvest row crops like corn and soybeans. The technology — two years in development — was originally developed in a lab using computer simulation with the assistance of Jaybridge Robotics (**www.jaybridge.com**), a robotic software company founded by graduates of MIT and Carnegie Mellon. The bottom line is that farmers can now spend less time planting and harvesting, and more time making them tasty corn squeezins. **SV**

An unmanned John Deere operated by the Kinze autonomous system.

GEERHEAD

by David Geer

Contact the author at geercom@windstream.net

Lingodroids Form Their Own Language

to Name, Describe, and Share Places With Each Other in a Useful Manner

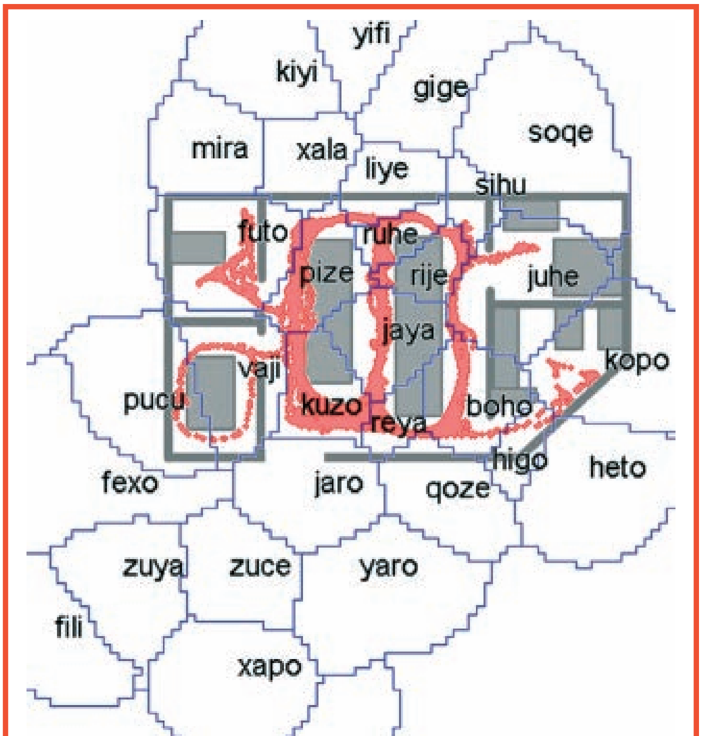
Lingodroids research enables two robots to discover and map new locations, name those locations with distinct ring tone sounds, and then share the maps. Once an environment and all the locations or areas within it have been named, one robot can tell another robot that has not been there how to get there. The Lingodroids can travel to areas they have not been to before by noting the positions relative to places they have been.

About Lingodroids

The robots create their own naming system on the fly as they explore an environment. This is much easier than requiring them to use languages created by human beings. Once a robot has randomly named a location, it knows that the word it has created represents that location and is associated with how to get there, where it is, and where the robot is relative to the location.

Lingodroids are programmed robots based on the Pioneer3 DX platform available from Mobile Robots. The Lingodroids also sport a 360 degree panoramic camera setup, a laser range finder, and a microphone and speaker system. The robots see using a RatSLAM vision-based SLAM (Simultanteous Localization, Mapping, and Path Planning) system that uses images from the 360 degree camera to construct visual scene matches. The robots use odometric information from their wheel encoders to assist with distance information. The robots use the range finders to avoid objects and follow walls unhindered while exploring their current environment.

"The robots use the microphone and speakers to establish a shared attention. If they can hear each other, then they know they are close to each other," says Dr. Ruth Schulz, lead researcher, University of Queensland, Australia. The robots use a wireless network to communicate other information between themselves such as words in a text string to enable their interactions.

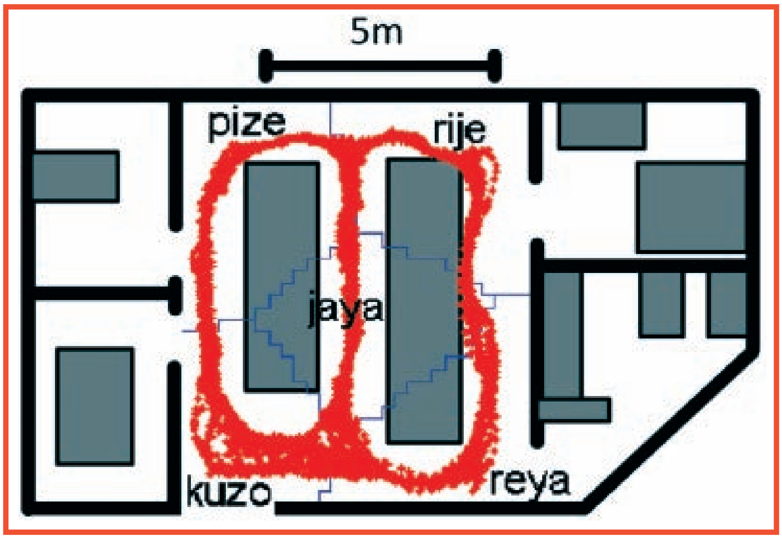


This is a map of the different areas the Lingodroid robots (which are based on the Pioneer3 DX platform) have seen, named with their own newly created vocabulary, and recorded.

Lingodroid Language

"When a Lingodroid decides it needs to create a new word, it invents the word by combining random syllables. These syllables are translated into the beeps the robots share with each other or remain as text only for wireless

This is a map of some of the areas that the two Lingodroid robots have seen and shared names of. The red scribbling represents a path around objects that the robots can take or have taken to reach each named area or destination.



communication," explains Dr. Schulz. The beeps are used in language games.

The robots learn what the words mean through these games. Each language game establishes certain concepts or topics about the word at hand. "The Lingodroids learn an instance of a word in a single experience in the same way that a child does. They refine what they understand about the word in later interactions in which they use the word. New words compete with old words to determine which is the best word for a given situation," says Dr. Schulz.

Locations are a great place for robots to start creating their own language because the meanings of the words are concrete and specific, and the robots can easily associate the words with things they understand.

The Games Robots Play

The two Lingodroids start by playing "Where are we?" games. In these interactions, if the robots cross into an area that they have not named yet, one or the other of them will invent a word for it. "The robot then communicates the word to the other robot upon meeting, defining the name of the location," says Dr. Schulz. These words are known as toponyms. "Topo means place and nym means name," Dr. Schulz explains.

The next games the robots play are "How far?" and "Which direction?" These communications enable the robots to develop relationship words that are much like our human English prepositions. "The resulting language makes use of location, distance, and direction words which enable the robots to refer to new places based on their relationship to known locations," Dr. Schulz continues.

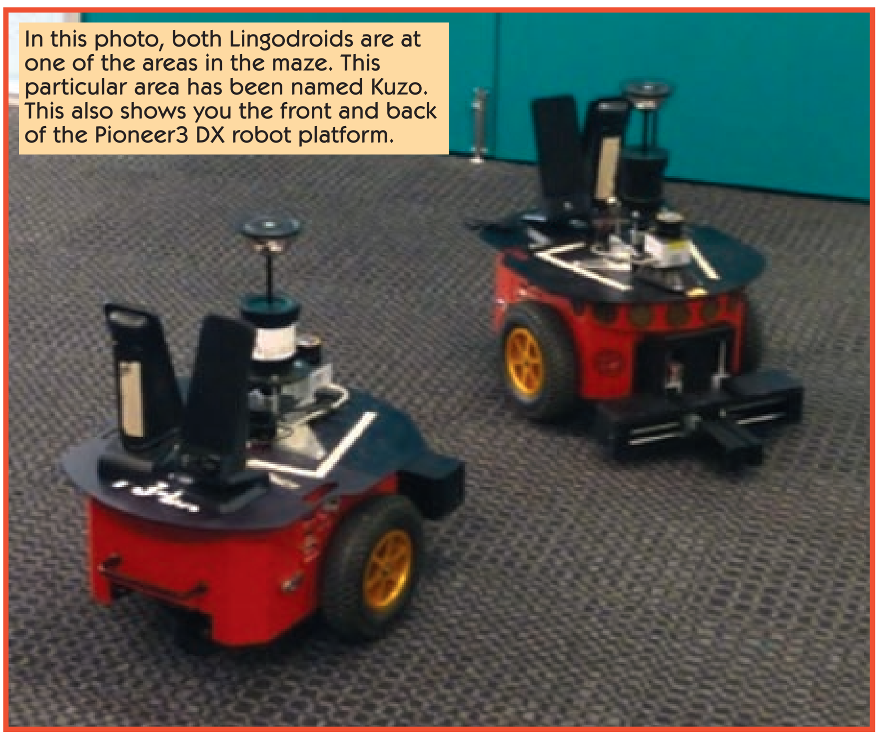
These are powerful and generative languages because they enable the robots to refer to new places they haven't been to yet or even places they imagine beyond the edges of their known environment.

The robots are able to associate the new or unknown places with positions within the global framework of the world they know and have currently mapped out. Each unvisited but named new or unknown location or place is looked at as a new experience waiting for the robot that it can find based on its known map. "We call these new experiences pseudo-experiences. We use the term pseudo as the places are not locations that have been directly experienced by the robots but rather have been indirectly referred to through language," says Dr. Schulz.

Lingodroid Components

The Lingodroids each have numerous components that add to their linguistic and navigational capabilities. The mapping is performed by the SLAM system referred to earlier. This system constructs the maps of the robots' experiences based on the paths they took to discover each new location in their world.

The robots build these cognitive maps based on the personal experiences they have in their world. "The maps are private and unique for each robot. This system of



In this photo, both Lingodroids are at one of the areas in the maze. This particular area has been named Kuzo. This also shows you the front and back of the Pioneer3 DX robot platform.

Note the speakers in this side angle shot of Lingodroids A and B, which are used to make their communications with each other audible.

mapping was inspired by the mammalian hippocampus and was developed at the University of Queensland and the Queensland University of Technology by a team including Dr. Michael Milford and Professor Gordon Wyeth," says Dr. Schulz.

The researchers used a distributed lexicon table to associate the concepts and words that the robots create and use. "Instances of each concept are experienced in an interaction. The relevant information is extracted for the type of concept (the current location for a toponym, the calculated offset between two locations for distance). That information is associated with the word used in the interaction," remarks Dr. Schulz.

Word production and comprehension processes are then applied. The shared language that the robots develop allows them to talk about places in their world without reading each others minds by direct map sharing.

Research Goals

"Our eventual target is to develop robots that can communicate meaningfully and effectively with humans, as well as other robots. This research takes us one step closer to robots that are more useful to humans in domestic situations, as well as having a robot in your home that you can interact with naturally," says Dr. Schulz.

It should be feasible some day to have a domestic robot come out of its shipping box with a basic understanding of a language, and then learn what parts of a house correspond to the word kitchen (for example) and so on for other parts of the house, as well as what you want the robot to do in that room (such as to clean it up).

Note the bot's on-off switch on the side toward the back.

Conclusion

Once you realize the goal and how much closer the research brings us to it, it is easy to see the value in this location language creation experiment. What seems primitive from the viewpoint of a usable household product is advanced from the perspective of what has been accomplished before.

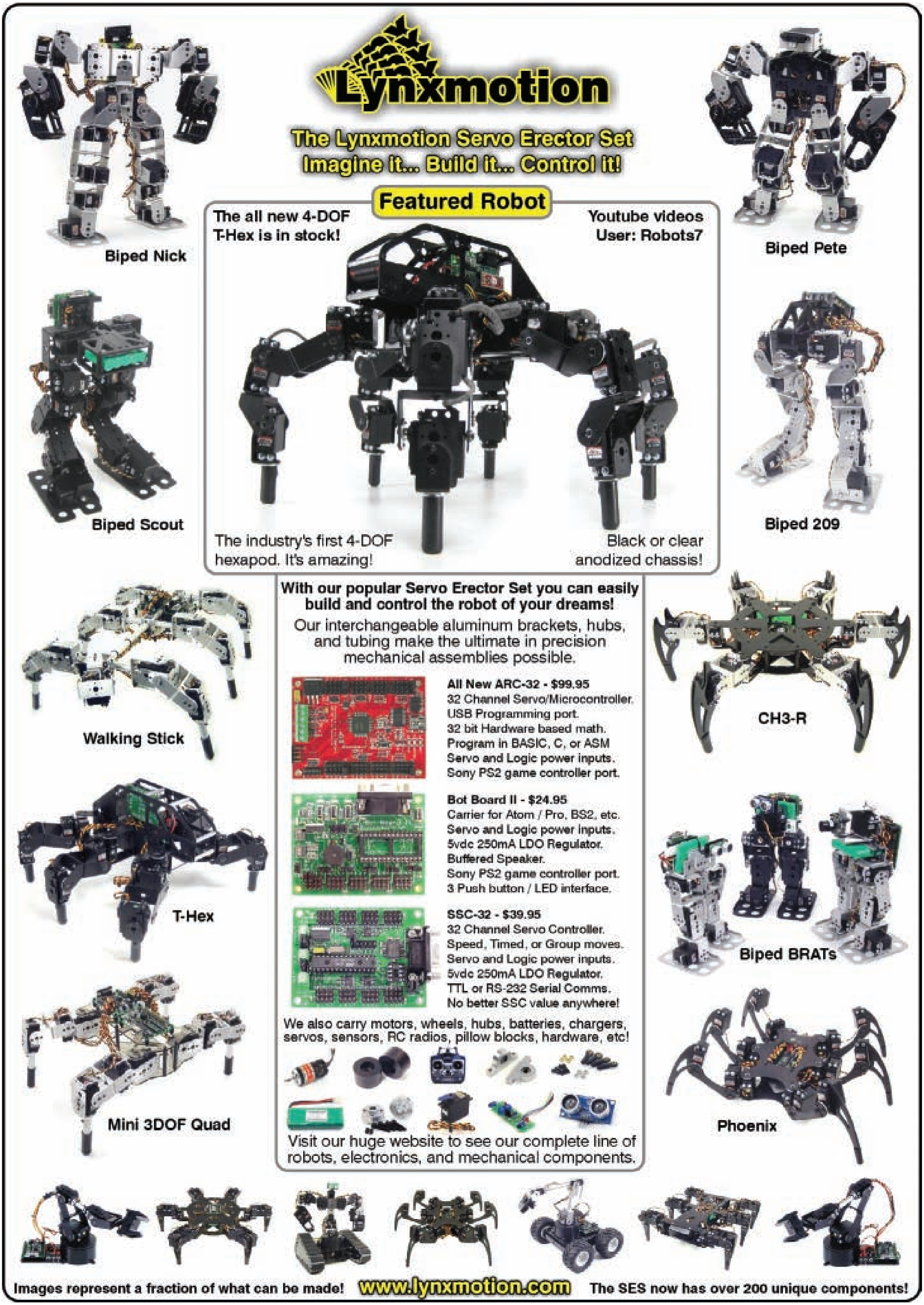
According to the Emerging Technologies Hype Cycle 2010 from Gartner, mainstream adoption of mobile robots is still more than 10 years out. But work like that of Dr. Schulz on Lingodroids will contribute to mobile robots being highly capable and useful once they do hit the mainstream. SV

Resources

Home page of Dr. Ruth Schulz of The University of Queensland, Australia; Lingodroids principal researcher
http://itee.uq.edu.au/~ruth

The University of Queensland School of Information Technology and Electrical Engineering
www.itee.uq.edu.au

The Pioneer3 robot platform
www.mobilerobots.com/researchrobots/researchrobots/pioneerp3dx.aspx


Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Featured Robot
The all new 4-DOF T-Hex is in stock!
Youtube videos User: Robots7
The industry's first 4-DOF hexapod. It's amazing!
Black or clear anodized chassis!
Biped Nick
Biped Pete
Biped Scout
Biped 209
Walking Stick
CH3-R
T-Hex
Biped BRATs
Mini 3DOF Quad
Phoenix
With our popular Servo Erector Set you can easily build and control the robot of your dreams!
Our interchangeable aluminum brackets, hubs, and tubing make the ultimate in precision mechanical assemblies possible.
All New ARC-32 - $99.95
32 Channel Servo/Microcontroller.
USB Programming port.
32 bit Hardware based math.
Program in BASIC, C, or ASM
Servo and Logic power inputs.
Sony PS2 game controller port.
Bot Board II - $24.95
Carrier for Atom / Pro, BS2, etc.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
Buffered Speaker.
Sony PS2 game controller port.
3 Push button / LED interface.
SSC-32 - $39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
No better SSC value anywhere!
We also carry motors, wheels, hubs, batteries, chargers, servos, sensors, RC radios, pillow blocks, hardware, etc!
Visit our huge website to see our complete line of robots, electronics, and mechanical components.
Images represent a fraction of what can be made!
www.lynxmotion.com
The SES now has over 200 unique components!

Our resident expert on all things robotic is merely an email away.

roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

by

Dennis Clark

ASK MR. ROBOTO

The summer flew by, fall is flying by as well, and competitions loom like the sword of Damocles over our heads. Anyway, I have a couple of interesting questions, some commentary, and a discovery to share this month! Without further adieu, I shall move into the meat of my column ...

Q. I would like some clarification and help, but I'll be as brief as possible. I have seen a few different Polaroid sonar projects and yours is one of them. The original Polaroid project that I saw was from *Ciarcia's Circuit Cellar* by Steve Ciarcia. I may be using a different board all together, but how can you tell other than looking at the pictures and compare? The ones I have looked at are very much like what you and Steve are showing in your articles.

I have the parts for the Polaroid 660 and the SUN 660 and One Step, which I would really like to get working. However, I see discrepancies between each one of these hacks and projects, and I have not been able to make any progress.

In your hack from "Polaroid Sonar hack.html" on **www.seattlerobotics.org/encoder/index.php**, I am not clear as to where you have the "-" and the "+" terminals of the "100 µf" capacitor attached to; the picture is very hard to see.

In that picture, you are showing the pin-outs on the Polaroid 660 board in **Table 1**. I am also showing what Steve Ciarcia has in his *Build the SonarTape* article (**www.micromint.com/app_notes/ti01_data.htm**) for the "J1" connector in **Table 2**.

1	2	3	4	5	6	7	8
Vss	BLINK	BINH (PWR)	INIT	FIL	OSC	Echo	Transmitter Base

Table 1.

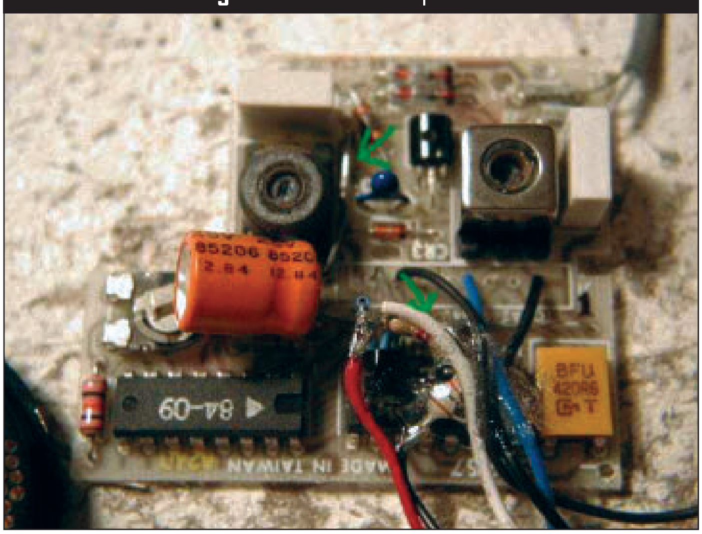
1	2	3	4	5	6	7	8
V+	ECHO	OSC		INIT	BLNH	BLNK	GND

Table 2.

So, as you can see I am left puzzled with yet another mystery.

— Givi (via internet)

Figure 1. 660 One Step sonar.



A. Let me answer your questions as best I can, in the order you gave them. I think that we can clear this right up. The SRS *Encoder* article you refer to originated from my website where I posted the results of weeks (maybe months) of experiments I did with at least four different types of old Polaroid cameras that had sonar boards in them. In those days, there weren't a lot of inexpensive options for sensors in general, and sonar in particular (has it really been that long?). I scoured the Web, called and talked to Polaroid folks, and hunted down technical documents to get this information, but still it was incomplete and I had to experiment a bit when things didn't work. My notes were

not intended to be a tutorial, so there are perhaps (ahem) a few holes ...

Your first question on "how to tell the boards apart" is right on. You have to look at them to tell. There is no information other than our posted pictures. Fortunately, they do look distinct enough that examination works fine. The very oldest of the Polaroid sonar boards is the "One Step." That board looks even older with its odd configuration. This board is the hardest to hack because you need to cut traces to keep it from doing things that worked fine in the camera (working the focus), but that you don't want to happen in your robot. We hijack some of these pins for our own use. The Sun 660 and 660 One Step look similar, but upon inspection, are very different. The most obvious difference is the existence of "C63," a trimmer cap on the newer Sun 660 boards (see **Figure 3**).

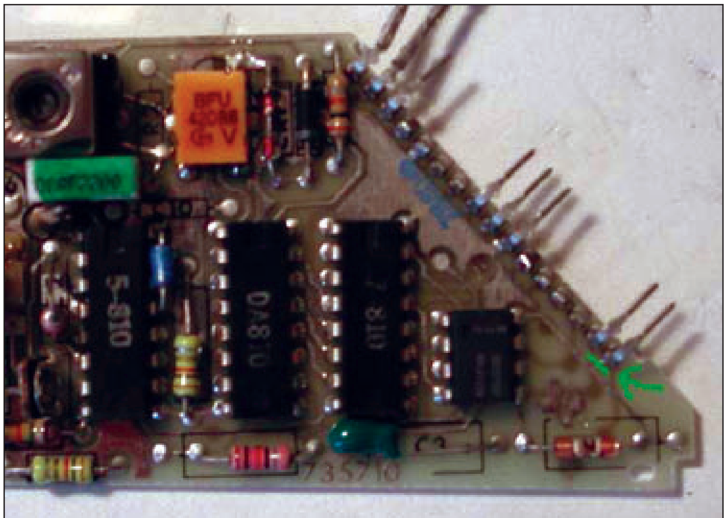


Figure 2. Polaroid One Step sonar board.

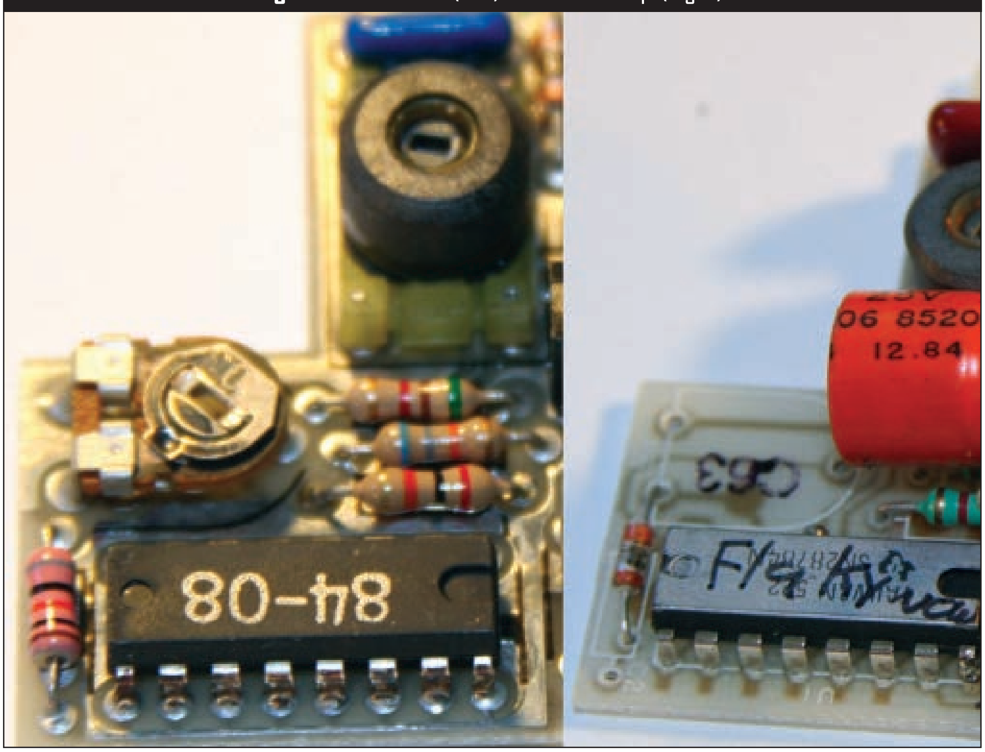
My notes are clear on how to install the reservoir capacitor on these two boards, except perhaps about where the - and + leads go. So, let me answer that question next. My notes clearly label these leads, but on the older 660 One Step it may not be clear how to attach them. Look at **Figure 4**; the + side of this large capacitor is on the right. If you have this old 660 One Step board oriented as I show, you will see a bare wire on the top side of the board, running horizontally under the black choke. Solder the + lead of the cap to this wire. Somewhat more difficult to see from a picture is the bare wire on the top side of the board running vertically alongside the connector. This is a ground wire; solder the - side of the capacitor to this wire.

Because the connector on this board is not something that easily matches any connector that I ever had, I removed it so that I could solder wires directly to the board to handle the power switching and signal inputs and outputs that I wanted.

This leads us to the third question; the one about "J1." Because there really isn't any documentation about this board — at least none available over a decade after the camera ceased production — we are left on our own to determine how to label the connector. I used the label orientation that Polaroid used on its 6500 sonar experimenter boards (also obsolete shortly after I discovered them). It wasn't a perfect match; the 6500 J1 had more pins, for instance, but the pin description matched up to pin 8. I don't know what decided *Circuit Cellar's* pin orientation for J1. If you look at the two, you'll see that the pins are simply mirrored. My pin 8 is V+ which has the transistor attached to it so that the microcontroller can turn the power on and off to the board at will to reset it. By the way, the collector is tied to pin 8, not the base of the transistor. I use a PNP transistor to handle this because the large capacitor I use as an energy reservoir can be charged easily with the TO92 transistor when the board is powered on. Its large value can supply the high current spike caused when the board triggers its sonar pulse without disturbing the microcontroller.

I wrote about my Polaroid sonar hack discoveries over 10 years ago. It hasn't been until the last couple of years

Figure 3. Sun 660 (left); 660 One Step (right).



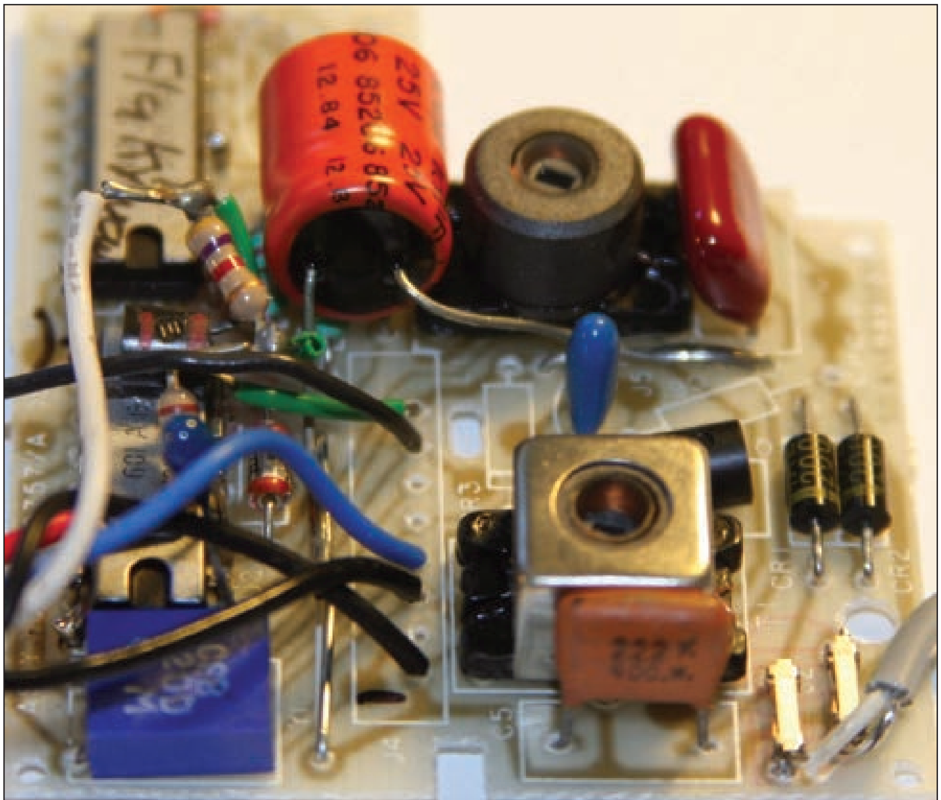


Figure 4. Close-up of 660 One Step sonar board.

that any other company has come out with long range sonar boards, which makes these hacks still useful today as low cost, long range (out to about five meters or so) sonar sensors. You can still get these cameras cheap on eBay. Do NOT pay more than $5 for these cameras! Some people think they are antiques and worth their weight in gold; don't bother with those sales.

Q. I want to get into PIC programming. I found circuits on the internet that I can build into a cheap programmer for the PIC16F84. Can you help me with them?

— Robert (via email)

A. I hear from people trying to make the NOPPP (No Parts PIC Programmer) every once in a while, but rather less these days. This is *really old* information that — while it works — is really not worth your time and frustration. It requires a custom, bit-banged programmer interface and software that was written long ago, and isn't really supported any more. You would do yourself a favor to save up a little money and get a genuine Microchip PICkit 3 for about $50 US which will allow you to program and even debug many of the Microchip parts. If you want even cheaper, you can get a Cytron PICkit 2 clone for about $20 US or an Olimex PICkit 3 clone for about $30 US. All of these are MPLAB compatible and will work with a variety of PIC processors. Places to look for MPLAB compatible Microchip or compatible programmers/debuggers are:

www.digikey.com
www.botshop.com
www.olimex.com
www.sparkfun.com

There are more; these just came off the top of my head.

Now, about that PIC16F84 (or the even older 16C84) ... This is a soapbox of mine. Don't bother with those old parts if you are a hobbyist. Even if you have a box of these things, you should just recycle them and move on. That part is over 10 years old and technology has marched forth! Look at the 18F or 18FJ parts in the eight-bit processors and seriously consider moving up to the 24FJ or 24H parts. These latter are 3.3V parts (which matches nicely with all of the new 3.3V sensors) and are *MUCH* faster. In some cases, they cheaper than those old 16F parts! If you are on a budget, choose a PIC that the PICkit 3 can program (which is most of them) and you'll be glad you moved on.

The great advantage of the 24F parts is that Microchip has the C24 C compiler for FREE on their website. One great experimenter board is the MICROSTICK DSPIC33F/PIC24H. This has a 24H processor (and a dsPIC too) and has the programmer built in. Just plug it in your USB and run MPLAB to start developing. Cost? About $25 US. It's the board, programmer, microcontroller, and everything else in a package about the size of a stick of gum (more or less). So, leave those old programmers and PIC parts behind and join the 21st century. Really, you'll be glad you did!

That's it for questions, so I want to move on to something that I discovered recently — the Digilent (**www.digilentinc.com**) ChipKit UNO32 boards. You have seen the ads in *SERVO* and other places for the ChipKit UNO32 that say they are "Arduino compatible." If you are like me, you just said, "Whatever. Arduino is more than just something that takes I/O shields." I really like the Arduino environment, attitude, and user group. Well, I'm here to say "Pay attention! They *mean* Arduino compatible!"

I ran across a couple of guys not long ago (Mark Sproul and Rick Anderson of Rutgers University) while at a conference. They had been hacking selflessly to write upgrades to the Arduino IDE that expanded it to include the UNO32 series of boards that run the Microchip PIC32 (32-bit) processor at 80 MHz. I've gotten one of these boards along with an I/O expander board for the UNO32 and have worked with the new MPIDE Arduino platform. Wow.

It looks like an Arduino, programs like an Arduino, and even runs the standard Arduino programs. The port isn't complete at this moment, but by the time you read this, it may well be. This is an Arduino on steroids. The UNO32 has 128K Flash and 16K of RAM. You can do a lot with that!

Rumor has it that the Arduino developer crew is going to accept most of the MPIDE expansions into the next official Arduino IDE release. Mark and Rick believe this will open the door for others to port different hardware to the Arduino stable more easily. Digilent has even stayed within the fully open source Arduino mindset and has made the ChipKit UNO32 board designs and bootloader fully available for anyone else to create their own UNO32 designs. So, if you ever wished that your Arduino could be faster, your wish has been granted.

Running the typical demo programs from Arduino (ones not optimized to the AVR ATmega parts) will look the same as they do on a 16 MHz ATmega Arduino; they'll just run about 6-7 times faster (by my ad hoc timing experiments). I've even gotten the SparkFun color LCD shield and NKC color LCD shield programs to run.

There are a few things to look out for, and they all have to do with code written and optimized on the ATmega parts. If you translate them to C32 (Microchip's C compiler) format, they'll work on the UNO32 series of boards. (If there is interest in this, I can do a more in-depth article on the UNO32 boards in the future.) **Figure 5** and **Figure 6** are shots of the UNO32 and its basic I/O shield from Digilent. The basic I/O shield has the following stuff on it:

- 128x32 pixel OLED graphic display.
- I^2C temperature sensor.
- 256 Kbit I^2C EEPROM.
- I^2C daisychain connector.
- Four pushbuttons.
- Four slide switches.
- Eight discrete LEDs.
- Four open drain FET drivers.
- Analog potentiometer.

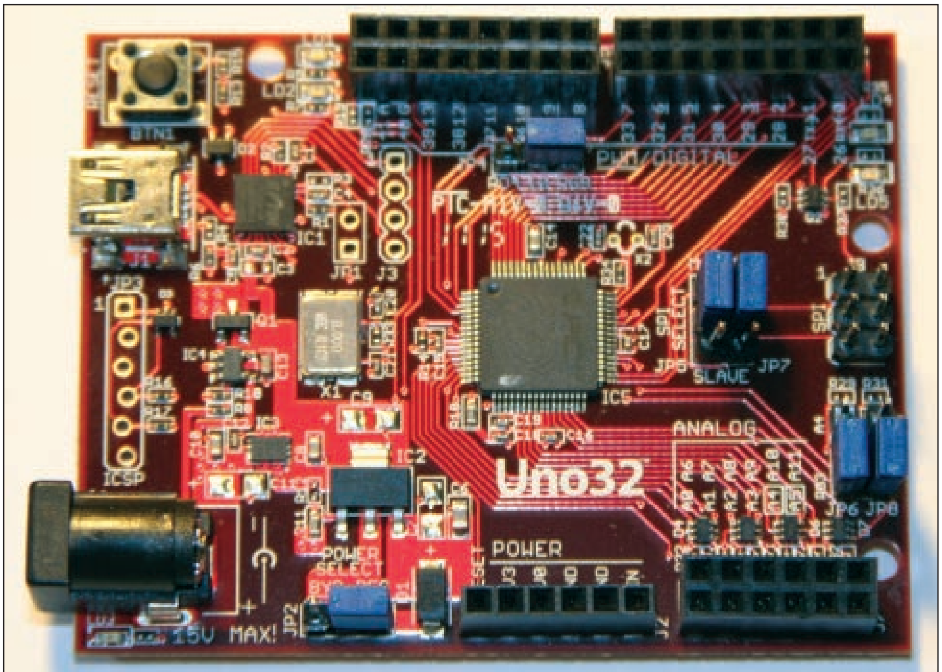
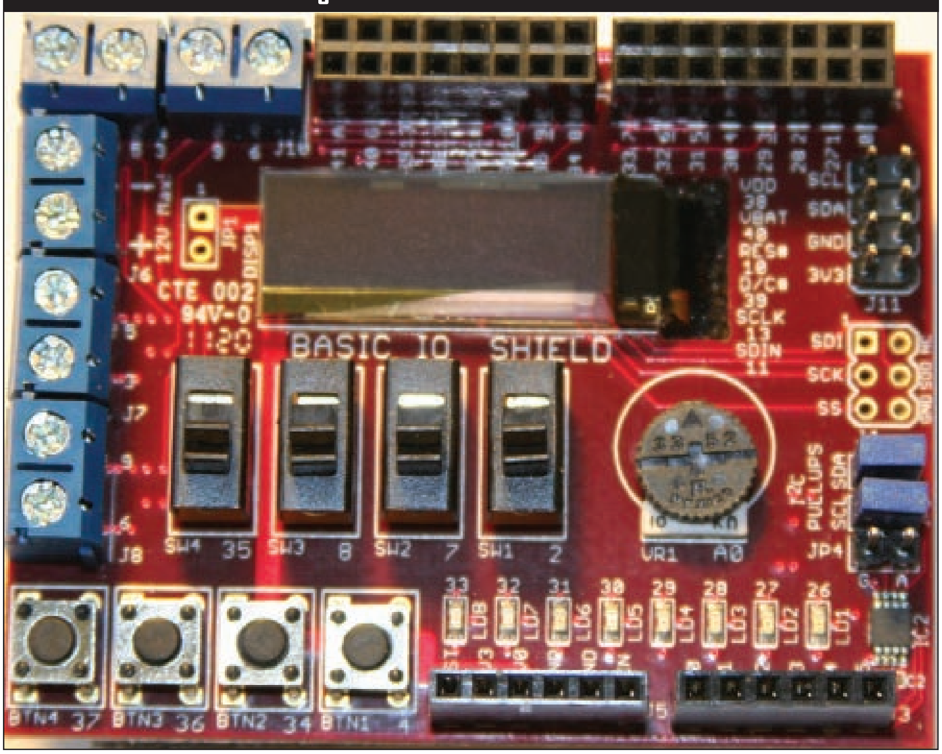


Figure 5. ChipKit UNO32 80 MHz Arduino board.

The four FET drivers are "low side" 20V, 3A drivers useful for controlling relays or single direction motors (a stepper, for instance). The OLED display is pretty cool too. The Arduino hackers are porting sketches to control these pieces; these scripts should be available soon.

Well, that's it for another Mr. Roboto this month. Keep on building robots, and if you have any questions about new ways to do, old ways to do things, or new things in general, drop me an email. I can be reached at roboto@servomagazine.com. **SV**

Figure 6. UNO32 basic I/O shield.



EVENTS Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**

— R. Steven Rainwater

NOVEMBER

6 **International Micro Robot Maze Contest**
Nagoya University, Japan
This competition is held along with the International Symposium on Micro Mechatronics and Human Science. Events include 1 cm Micro Robot Races, Teleoperated Mountain Climbing Micro Robots, the Autonomous Micro Robot Maze, and Micro Biped Locomotion Robots.
http://imd.eng.kagawa-u.ac.jp/maze

12 **AHRC Robot Rally**
Pinckneyville Community Center, Norcross, GA
This year's competition includes an Open Competition, R/C Qube Quest, Maze Solving, and a new six-part Polyathlon. There will also be mini Sumo exhibition matches.
www.botlanta.org/robot-rally

12 **DPRG RoboRama**
Museum of Nature and Science, Dallas, TX
This year's Dallas Personal Robotics Group fall RoboRama includes three indoor events: line following, square dance, and table top, as well as the outdoor RoboColumbus event.
www.dprg.org/competitions

12-13 **Canadian National Robot Games**
Ontario Science Centre, Ontario, Canada
Events include Sumo, line following, fire fighting, search and rescue, walkers, photovore, and more.
www.robotgames.ca

12-16 **Real World Robot Challenge**
Tsukuba Expo Center, Tsukuba, Japan
Autonomous robots compete in the real world, literally — they must navigate the Tsukuba streets and sidewalks, coexisting with humans, animals, and vehicles.
www.ntf.or.jp/challenge

18 **All Japan MicroMouse Contest**
Tsukuba, Japan
Super-speedy autonomous robots must solve and navigate a maze in the shortest time possible.
www.ntf.or.jp/mouse

18-20 **Texas BEST Competition**
Special Events Center, Garland, TX
Student teams build robots and compete in this regional competition, learning a variety of science, technology, engineering, and math skills along the way.
www.eng.unt.edu/texasbest

20 **Robocon**
Tokyo, Japan
Sixty-two technical colleges and 57 other schools nationwide participate in Robocon which culminates at this championship.
www.official-robocon.com

26-27 **Robotex**
Tallinn University of Technology, Tallinn, Estonia
Autonomous robots compete in line following, soccer, and FIRST LEGO league events.
www.robotex.ee

DECEMBER

1-4 **ROBOEXOTICA**
Vienna, Austria
Watch robots mixing cocktails, lighting cigarettes, and working toward other ground-breaking achievements in electronic cocktail culture.
www.roboexotica.org

10 **Robotic Arena**
Wroclaw, Poland
Autonomous robots compete in Sumo, mini Sumo, line following, and freestyle events.
http://lirec.ict.pwr.wroc.pl/~arena

27-29 **MindSpark**
College of Engineering Pune, India
Events include Micromouse, Dogfight, and Search and Destroy.
www.mind-spark.org

The Riverside Robotics Society, in association with the Robotics Society of Southern California, present the

RIVERSIDE ROBOT EXPO 2011

Come see robots of all sizes and shapes, including a WALL-E, an R2D2 robot, a "Lost In Space" robot, Robot Elvis, a Terminator robot, singing androids, a ghost hunting robot, robot races, mini Sumo battle bots, line following robots, art robots, and more. There will also be demonstrations and lectures in robot building, artificial intelligence, robot vision, speech recognition, and robotic ghost hunting with Albert Taylor.

Free Admission!
Robot Door Prizes!
Refreshments!

Saturday, November 19th
12 Noon to 4 PM

Riverside Woodcrest Community Library
16625 Krameria
Riverside, CA
(951) 789-7324

Robotics Showcase

NEW PRODUCTS

New 4-1/2 Digit Bench Digital Multimeter

Global Specialties has introduced a new digital multimeter. The PRO-1000 is a portable, bench type digital multimeter with a 4-1/2 digit LED display that can measure diodes, frequency, current, voltage, and resistance, as well as test continuity for a list price of $345.

Important features of the new PRO-1000 include:

- True RMS digital multimeter.
- Basic DC voltage accuracy of 0.05%.
- Basic AC voltage accuracy of 0.75%.
- Overload protection on all ranges.
- AC/DC voltage measurement up to 1,000V DC and 750V AC.
- AC/DC current measurement up to 10A.
- Resistance measurement up to 20 MΩ.
- Frequency measurement up to 200 kHz.
- Audible continuity test.
- Diode test voltage, 2V.
- 110-120V AC @ 50 Hz/210-220V AC @ 50 Hz input voltage.
- Weighs only 4.4 lbs (approx 2.0 kg).
- Comes complete with user's manual, one set of test leads, power cord, and spare fuse.

For further information, please contact:

Global Specialties, LLC	22820 Savi Ranch Parkway Yorba Linda, CA 92887 Tel: **800 • 572 • 1028** Website: **www.globalspecialties.com**

The X4 AC Plus, Four-Channel Multicharger with AC Power Supply

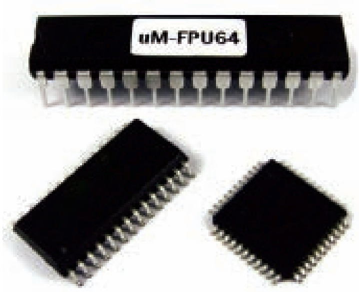
Hitec announces a brand new four-channel multicharger plus AC power supply called the X4 AC Plus. The X4 is equipped with a built-in AC power supply and has a four-port, computer controlled capability with a built-in 22 amp power supply which can be powered by a standard 120 volt AC household current. Simple to use and reliable, this compact charger with battery management and integral balancer features four identical and independent 50 watt power outputs, for a total output power of 200 watts. Each port of the X4 AC Plus can handle up to a 6S Lithium pack, 15 NiCd/NiMH cells, or 6-12 volt lead-acid batteries. Four individual balancing ports eliminate the need for a separate balancer when charging Lithium batteries. Its twin fan cooling system with an internal sensor for controlling fan speed provides efficiency and safety. The X4 can accept a variety of power inputs, including the option of attaching it to a 12 volt car battery or a 11–15 volt (20 amp minimum) DC power supply.

The X4 AC Plus charger offers a complete charging solution for novice and expert RC enthusiasts alike. The multi-tasking versatility of this new charger makes it a perfect fit for nitro and electric modelers, regardless of whether it drives, hovers, floats, or flys.

For further information, please contact:

Hitec	12115 Paine St. Poway, CA 92064 **858 • 748 • 6948** Fax: **858 • 748 • 1767** Website: **www.hitecrcd.com**

64-bit Floating Point Coprocessor



Micromega Corporation announces the release of the uM-FPU64 floating point coprocessor chip. The uM-FPU64 extends Micromega's family of coprocessors to provide support for IEEE 754 compatible 64-bit floating point and integer calculations, expanded digital I/O and analog input capabilities, and support for local peripheral devices. The uM-FPU64 can be interfaced to a wide range of popular microcontrollers to provide extensive floating point capabilities, and optionally control a subsystem of local peripherals. It can also be configured as a stand-alone microcontroller for embedded applications.

The precision required for GPS navigational calculations and the transformation of data from MEMS-based sensors can easily exceed the capabilities of 32-bit floating point numbers. The uM-FPU64 coprocessor — with support for both 64-bit and 32-bit floating point numbers — provides the added precision needed for these

demanding applications, and can offload the floating point calculations from the microcontroller.

The uM-FPU64 is compatible with the instruction set of Micromega's popular uM-FPU V3.1 32-bit floating point coprocessor. Advanced instructions are provided for fast data transfer, matrix operations, FFT calculations, serial input/output, NMEA sentence parsing, string handling, digital input/output, analog input, and control of local devices.

Local device support includes: RAM, 1-Wire, I^{2}C, SPI, UART, counter, servo controller, and LCD devices. A built-in real time clock and foreground/background processing is also provided. The uM-FPU64 can act as a complete subsystem controller for sensor networks, robotic subsystems, IMUs, and other applications.

The uM-FPU64 IDE (Integrated Development Environment) makes it easy to create, debug, and test code. Code can be written in the IDE's high level language or in assembler, then compiled to generate code targeted for one of the many microcontrollers and compilers supported, or it can be stored internally in Flash memory. The IDE provides support for editing code, compiling, tracing code execution, setting breakpoints, examining registers, and programming user-defined functions in Flash memory.

The uM-FPU64 chip is RoHS compliant and has an operating voltage of 3.3V, with 5V tolerant SPI and I^{2}C interfaces. SPI interface speeds up to 15 MHz and I^{2}C interface speeds up to 400 kHz are supported. The chip is available in PDIP-28, SOIC-28, or TQFP-44 packages. The single unit price is $24.95 with volume discounts available.

For further information, please contact:

Micromega Corporation

1664 St. Lawrence Ave.
Kingston, ON K7L 4V1
Tel: **613 • 547 • 5193**
Website: **www.micromegacorp.com**

Rover Kit Provides Ideal Platform for AUV Competitions

A Two-in-One Super Rover Kit that provides everything needed to build either a 4x4 Super Crawler or an alternate 6x6 Rover Chassis is now available for purchase from MINDS-i.

Both configurations are designed for use in all-terrain Autonomous Unmanned Vehicle (AUV) competitions that are based on the DARPA Grand Challenge. Such competitions often require the development and programming of robots which navigate through an outdoor course using GPS, avoiding obstacles with start and end points out of the line of site.

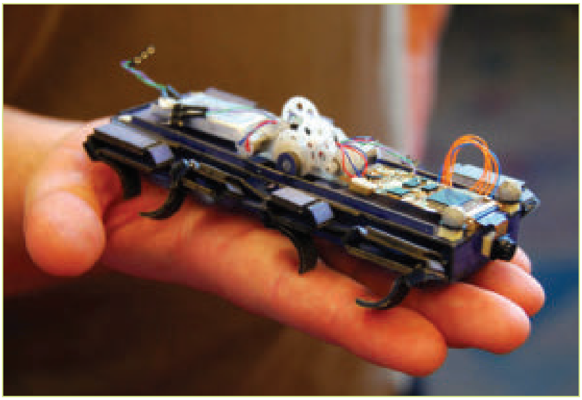
"AUV competitions such as Robo-Magellan are the most exciting and relevant because of their outdoor all-terrain unpredictable nature, which is a lot like reality," according to Mike Marzetta, MINDS-i inventor. "In contrast, many of today's more popular robot competitions unfortunately still take place on tabletops and gymnasium floors. I have never seen a photo of the surface of Mars with a flat paved surface waiting

Continued on page 74

The MINDS-i Super Rover, featuring six-wheel drive, six-wheel independent suspension, four-wheel steering, and 2.5 inches of suspension travel. One-of-two configurations offered in the Two-in-One Super Rover Kit.

The MINDS-i Super Crawler, featuring four-wheel drive, four-wheel steering, and advanced four-link suspension capable of 75 degrees of axle rotation. One-of-two configurations offered in the Two-in-One Super Rover Kit.

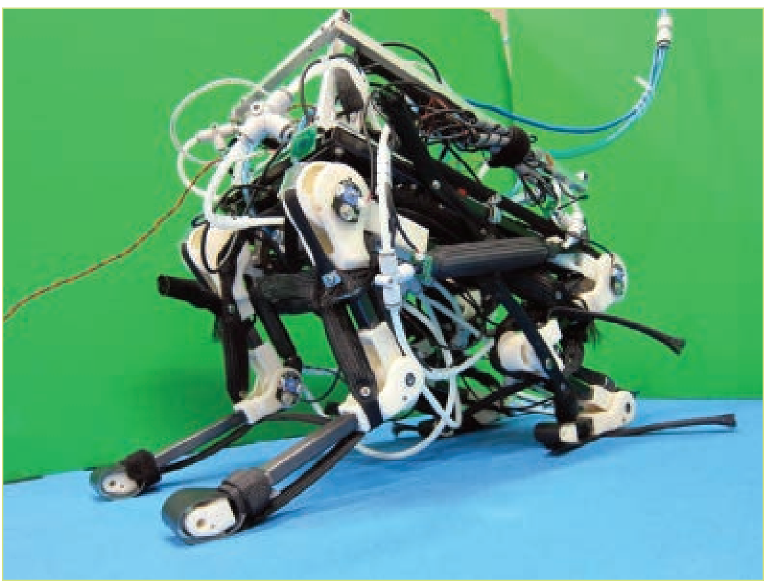
bots IN BRIEF



BIRDS AND THE BUGS

No matter how fancy and complicated we make robots, nature always seems to have us beat. For example, is there anything more capable, more efficient, and more indestructible than a cockroach? Of course not. Well, not yet, anyway. UC Berkeley's Biomimetic Millisystems Lab is trying to harness all the cleverness of birds and insects to create an entirely new generation of small robots with insect-like capabilities, and one of their most recent creations is called "Octoroach."

Octoroach has eight compliant legs and is small enough and light enough to rest comfortably in your hand. Batteries, sensors, and navigation are all completely integrated. Eventually, Octoroach is destined for the military to provide that last 100 meters of vital close-up surveillance. Of course, if 100 meters ends up being too far, you can just drop off your robo-roaches using robo-birds like BOLT — which stands for "Bipedal Ornithopter for Locomotion Transitioning." It's got a pair of little legs under its wings, and it can skitter around on the ground and over obstacles saving energy by not flying unless it has to. Berkeley is also working on a second ornithopter called iBird which is capable of flying towards a reflective target completely autonomously.



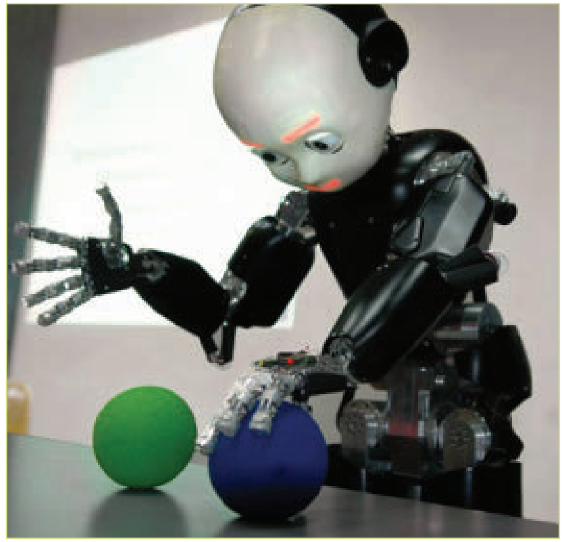
POGO PIGO

PIGORASS is a pneumatically-driven quadruped robot developed by Yasunori Yamada, Satoshi Nishikawa, Kazuya Shida, and Yasuo Kuniyoshi at the ISI (Intelligent Systems and Informatics) Lab — the same lab that brought us the jumping robot Mowgli and the running Athlete Robot. PIGORASS: its skeleton (made of ABS resin and carbon fiber reinforced plastic), 10 artificial pneumatic muscles, and 10 passive spring muscles weigh only 4 kg (8.8 lbs). Its total body length is only 35 cm (13?) long. The artificial muscles are driven by an external air compressor, and pressure sensors and potentiometers replicate how real muscles sense their length and tension.

What's noteworthy about PIGORASS is that its movements are not programmed in advance (as you would with robots using conventional motors). They emerge from the structure of its body and fluctuating signals from each muscle's neural oscillator using what is called a spinobulbar model. The neurons that individually control each muscle can fire in pairs (alternating between front and hind legs) to perform a kind of gallop, or fire all at once to produce a jumping motion.

bots IN BRIEF

CARRYING A TORCH



Dr. James Law, a researcher at the Department of Computer Science at Aberystwyth University, has an absolutely fantastic idea: He's nominated the iCub robot to carry the Olympic Torch as part of the 2012 Olympic Games which will be held in London, England starting next summer.

Dr. Law is proposing that iCub be included in the torch-carrying relay in honor of the 100th anniversary of the birth of Alan Turing — one of the guys who arguably invented the computer and whose test for artificial intelligence robots are continually striving to pass.

The only problem with this idea is that the nomination rules specify that all nominees have to be at least 12 years of age which means that iCub wouldn't technically qualify. On the upside, however, nowhere does it say that nominess have to be human, so maybe iCub has a shot at this after all.

GETTING AMPED UP

OLogic is launching the Automated Music Personality (A.M.P.) — a two-wheeled, self balancing "robotic" music player that can be operated using a Smartphone (only the Android for now). The company will be able to ship it for $400 or less, which is an incredible low price for a self-balancing robot.

Self-funded to-date via consulting revenue, OLogic is looking for funding to help build efficient distribution channels and bring the A.M.P. to the market.

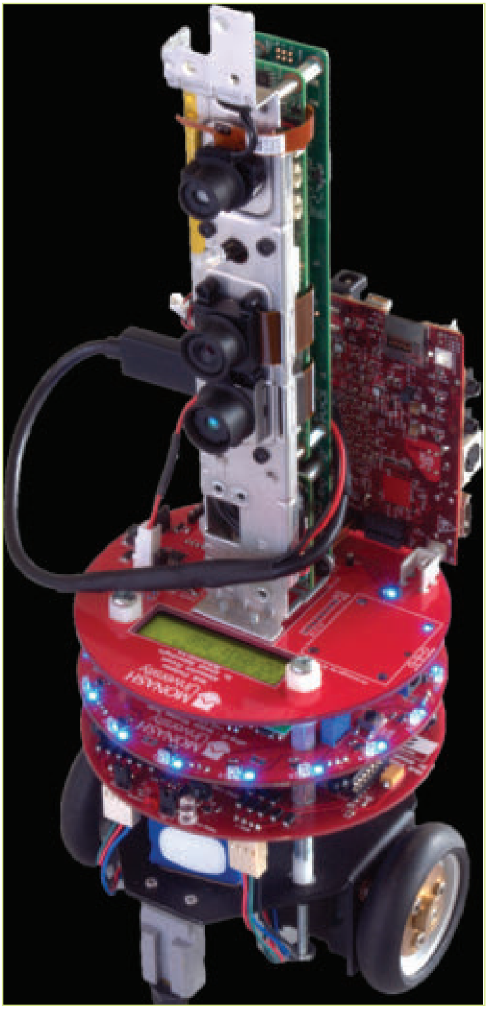
A.M.P. boasts a powerful audio system located in it chest, a cradle in the back to attach any type of music player, a cup holder, and several command buttons to control volume, for example. Using the proprietary remote control or your Android phone, you can make the A.M.P. follow you, make it dance, or stream music via Bluetooth.

Down the road, OLogic expects to add more features like the baility to replace the A.M.P. head with an iPad to offer an affordable telepresence robot.

EVOLTA-KUN DOES TRIPLE RUN

On October 23rd, our favorite little green guy participated in a week long Ironman Triathlon in Hawaii. Humans only get a day, but this trio of competitiors swam for 2.4 miles, rode a bike 112 miles, then ran 26.2 miles. During that time, the Evolta-kuns only stopped for recharging their nickel-metal-hydride batteries.

The three versions of the robot included one that swims, another that pedals a bicycle, and a third that runs in a hamster wheel contraption.



THE EYEBUGS HAVE IT

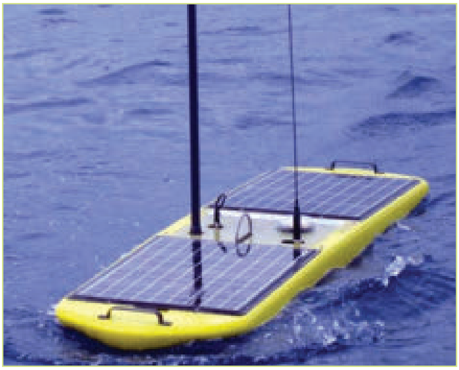
As part of his final year project, Monash engineering student, Nick D'Ademo has designed an experimental robot called "eyeBug," under the supervision of Dr. Ahmet Sekercioglu from the Department of Electrical and Computer Systems Engineering.

"When I gave Nick the specification, I had no idea what a beautifully flexible design he would create. We wanted a design that would be cheap to build, modular to be open for innovation and creativity, and finally something that would be visually appealing for other students," Dr Sekercioglu explained.

"Design is an integral part of our engineering students' curriculum where they are given the challenge of taking an idea and developing it into a fully functioning system. It is extremely rewarding for us to see the students applying the principles we teach so successfully."

By studying the way robots form networks to communicate with each other, it is possible to gain indispensable knowledge that will be used to develop the next generation of smart mobile phones.

Dr Sekercioglu's postgraduate students have already started developing technologies to extend the robot's capabilities to create distributed smart camera networks. Eventually, swarms of eyeBugs equipped with artificial intelligence algorithms will be able to build digital 3D models of their environment by communicating and sharing what every individual eyeBug sees.



RIDIN' THE WAVES

Now even water robots are becoming more self-sufficient! Liquid Robotics' Wave Glider uses the ocean's wave energy to move itself along, so no batteries are needed. Former astronaut and Google employee Ed Lu, Chief of Innovative Applications, says that onboard sensors monitor salinity, plankton activity, pollution, or water-based catastrophes like oil spills. It moves at only one and a half knots, but can also run on solar power. Theoretically, it can go for years without a break, unless an evil sea creature decides otherwise.

POWER WALKING

Bulow BioTech Prosthetics has announced its distinction as an iWalk Certified Bionics Center. With this designation, Bulow BioTech will serve as one of the nation's first prosthetics providers to commercially offer the PowerFoot BiOM — the first robotic lower leg system to normalize metabolic efficiency and increase self-selected walking speed for amputees. The BiOM features advanced bionics technology developed by Dr. Hugh Herr, an MIT-based, world-renowned innovator and researcher in the fields of biomechanics, biological motion control, and augmentation technologies.

During the test phase of the PowerFoot BiOM, Bulow was one of the first clinics to outfit a civilian with this technology. The BiOM is the world's first bionic lower leg system that utilizes robotics to replicate muscles and tendons that replace the action of the foot, Achilles tendon, and calf muscle. Scientific research has proven that the BiOM normalizes level ground walking for lower-limb amputees by enabling them to walk at the same speed, and with the same metabolic energy as their peers with intact biological limbs. For more information, visit **www.bulowbiotech.com**.

KINECT WITH EDDIE

Microsoft knows that Kinect is a big deal for robotics enthusiasts of all kinds, so they've announced the availability of a new beta release of Microsoft Robotics Developer Studio (RDS) that incorporates the full Kinect SDK that was released back in June. This includes skeleton tracking, speech, and the raw Kinect data stream for creating 3D maps of your house (or anything else).

Besides the full-fledged Kinect integration, the other big news about RDS 4 is that for the first time, Microsoft has their own hardware reference platform designed to make it fast and easy (sort of) for consumers to get straight to programming without having to actually build a robot. Eddie (pictured here) features a round multi-level design that incorporates a Kinect sensor and laptop. Robotics Developer Studio 4 Beta is available for download for free, and includes a simulation environment to get you started without needing to buy any hardware at all. When you're ready to take the plunge, Eddie is also now available, directly from **Parallax.com**.

HOT CHICKS

In one of the latest studies in the growing field of animal-robot interaction, researchers have found that young quail chicks that interact with autonomous mobile robots have improved spatial abilities later in life — at least up to a point. The results not only show the potential impact of robots on animal development, but also provide insight into the factors that play important roles in that development

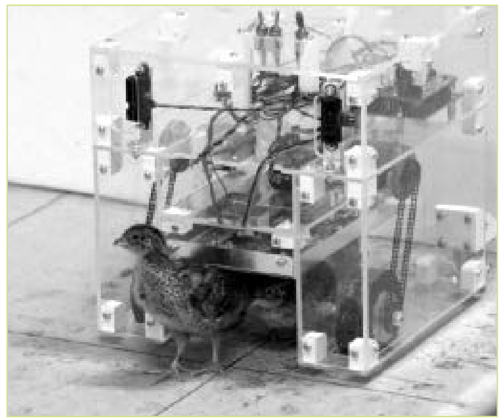


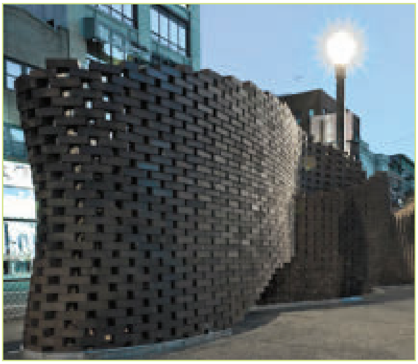
Image credit: E. de Margerie, et al. ©2011 IOP Publishing Ltd

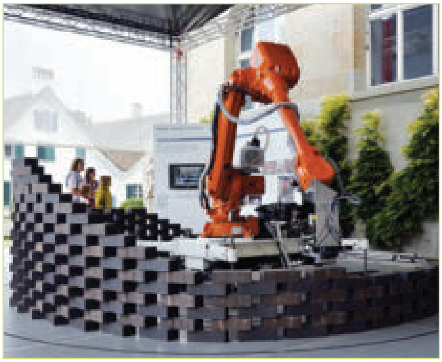
Although previous research has shown the importance of the mother hen in the development of quail chicks, using a mobile robot in place of the hen allowed the researchers to assess the impact with greater control. In their experiments, they divided 24 Japanese quail chicks into six groups of four. When the chicks were 36 hours old, the researchers began putting three of the groups in a cage with a heated mobile robot and the other three groups in a cage with the same heated robot, but with the locomotion feature disabled. Both of the cube-shaped robots had corridors between the wheels where the chicks could go to keep warm. The results showed that (five days after separation from the robots) the mobile robot improved the spatial learning and exploration skills of the chicks. However, the impact was probably transitory, as the other chicks seemed to catch up with their peers after another five days.

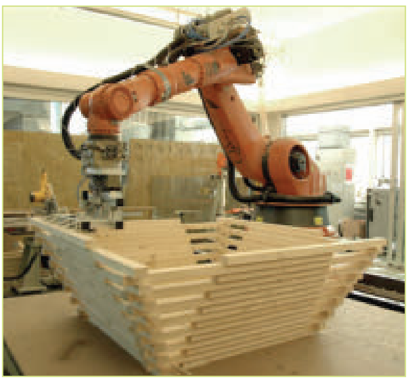
Although the effect was short-lived, the experiment shows how an autonomous robot can influence the behavioral development of animals. The researchers predict that more refined robots and prolonged meeting sessions may have stronger and more long-lasting effects on young animals.

SMARBO CLEANING UP

Toshiba has joined in the robot vacuum arena with the Smarbo. The VC-BR100 has 38 gyro and infrared sensors, a camera, and accelerometers to map out where it has been so it doesn't clean the same space. Cleaning is accomplished in parallel lines, and it can handle 1,076 sq ft in a 90 minute cycle. When finished or stopped, it returns to its starting point. All that technology will cost you, though. Sales started in Japan in October with a price of $1,175.







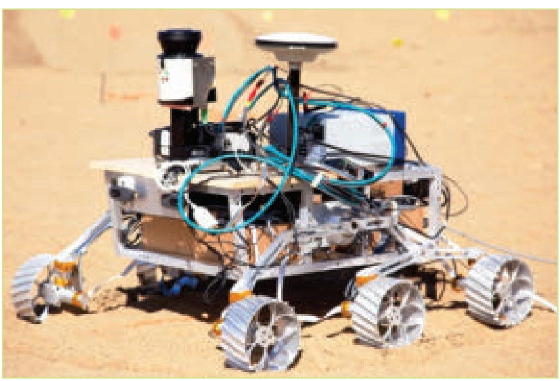
BUILDING BOTS

Fabio Gramazio and Matthias Kohler — both professors at ETH's Institute for Technology in Architecture — were among the first to use robots in architectural design. Since 2006, the duo has explored various manufacturing techniques, including both subtractive and additive fabrication — as well as a wide range of materials — to create astonishing structures built entirely by robots. The use of robots, combined with digital design tools, means a new aesthetic becomes possible, with novel shapes and patterns that would be nearly impossible to achieve without the automated machines, since industrial manipulators are extremely precise and good at repetition.

Using robots, the two ETH architects have fabricated intricate building parts out of wood, concrete, bricks, and foam, and have used these parts to build complex, beautiful installations in Zurich, London, Barcelona, New York, and other locations.

The idea of using robotic systems for reconfigurable spaces or "smart furniture" is not new. However, the way Gramazio and Kohler are using robots to actually build large environments is very innovative. Though their creations thus far have been limited in size, the architects are currently exploring the idea of applying robotic fabrication to the design and construction of high-rise buildings.

The architects are also collaborating with roboticists from the ECHORD project to give their robot more mobility. One idea is to use a base with tracks, and program the robot to recognize its position and surroundings. The biggest challenge is making sure the robot can handle construction tolerances and variations, adapting to changing conditions autonomously.



PUT A KAPVIK ON IT

Canada's Carleton University is working on Kapvik — a remote controlled micro-rover that can assist the more expensive Spirit Rover. So, the next time one gets mired down in Mars' mud, the six-wheeler can come to the rescue and then act as a scout the rest of the time.

The Canadian Space Agency is coordinating development of the rover, and partners include aerospace company MacDonald, Dettwiler, and Associates, as well as Toronto's Ryerson University which created a utility arm that will collect surface samples and perform trenching operations. Sensors planned for Kapvik include ultraviolet-visible spectrum, infrared imaging, and mapping tools to detect water and ice content. Kapviks will serve as low cost, adaptive rovers that will be remotely piloted and decrease the chances of losing more elaborate, expensive rovers to inhospitable terrain.

WEIGHT A MINUTE

The Intuitive Automata's Autom robotic weight loss coach is now available for pre-order on a dedicated "MyAutom" website. If you haven't been following the saga of Autom, it was first an MIT Media Lab robot with a significantly different look. Autom's developer at MIT, Cory Kidd, co-founded Intuitive Automata to help commercialize Autom based on the original MIT project, and it's starting to look like everything will be coming together within the next year.

Autom is designed to be exceptionally interactive, crunching data on your health, diet, and exercise regimen, and giving back friendly and constructive criticism. Studies have shown that people who use Autom stick with their diet and exercise routines for twice as long as people using more traditional weight loss methods

If this sounds good to you, you can be one of the first people to have this friendly little robot helping you out every day with a deposit of $195. This is not the final price, however. It's just the pre-order deposit. The final price is the $195 deposit plus a balance of $670 when the robot ships, for a total of $865.

HAVE YOU SEEN MY KEYBOARD?

Humans have the ability to look at a scene and immediately pick out important elements while ignoring everything else, mainly because we have brains that are awesome. Robots, in general, don't really work that way. They have to examine each and every pixel of what they're looking at and decide whether or not it's something they're interested in. So, for example, if you ask a robot go to find a computer keyboard, it's got to enter a room and methodically search every pixel-equivalent area until it finds what it's looking for. While the robot is likely to find its target in this manner, it'll take basically forever to do so.

However, the Cornell research group has been teaching robots to be able to rapidly break down a scene into general categories, and then recognize how some categories are related to others. In a general sense, this is the same thing we humans do. For example, if you enter a room and want to locate a computer keyboard, you first (subconsciously) identify the places that you don't need to pay attention to — like the walls, ceiling, and probably the floor. Next, if you don't immediately see the keyboard, you might take a closer look at the places where keyboards like to hang out, namely in close proximity to a computer.

The key to getting this to work goes back to the machine learning and adaptability system that Cornell has been working on. First, they showed a robot a series of about 50 different scenes, with most objects in the scene labeled. The robot read the labels and then remembered characteristics of the associated objects (like color, texture, and what other objects are nearby), so that it was then able to use those characteristics to categorize entirely new objects it's never experienced.

Armed with this knowledge and way of thinking, the robot was then able to enter a room it had never been in before in search of a keyboard, and "know" that since it spotted a monitor — and keyboards are often found near monitors — it should go check it out.

This ability to follow chains of reasoning to identify, categorize, and locate objects is still being developed, but giving robots the ability to understand context and use it to adapt to new things brings them that much closer to being those machines that take over all those household chores so we humans can do more important things.

Cool tidbits herein provided by Evan Ackerman at www.botjunkie.com, www.robotsnob.com, www.plasticpals.com, and other places.

COMBAT ZONE

www.servomagazine.com/index.php?/magazine/article/november2011_CombatZone

BUILD REPORT:

Rebuilding Apollyon, Again

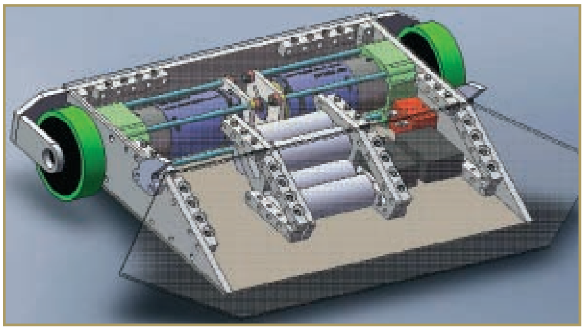
● by Mike Jeffries

Since the last build report was written, Apollyon has competed at Motorama. Apollyon took terminal damage there while achieving a 3-2 record, and yet again needed a complete rebuild. With the third build, I decided to focus on reducing complexity and using the internal space more effectively.

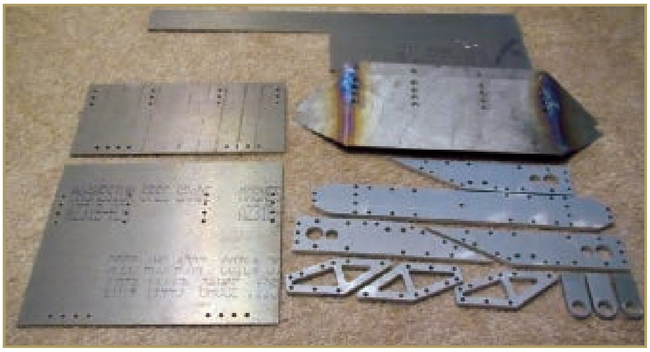
The major technique change used in this build was a switch from primarily tapped holes in the frame of the robot to a mostly waterjet cut chassis held together by a product called "Nutstrip" that is sold at **Kitbots.com**.

Nutstrip is a piece of square stock with a series of perpendicular holes drilled and tapped in it which allows parts to be easily bolted to it at 90 degree angles. Another benefit is that it is much less expensive and time-consuming to replace than an entire section of frame.

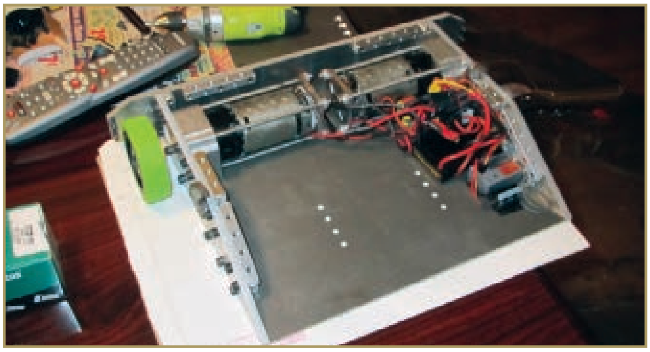
In addition to this construction methodology change, the materials had to be altered to provide better protection while not adding weight. Where the previous version used a 3/32" 4130 steel wedge, the new version uses 1/8" grade 5 titanium. Instead of aluminum and garolite for the top and bottom armor respectively, the new armor is



CAD model of the new Apollyon showing nutstrip and new chassis layout.



Parts as delivered from Team Whyachi. Materials are AZ31b magnesium, grade 5 titanium, and 7075 aluminum.



The entire electrical system has been crammed into one side of the chassis to keep wire lengths to a minimum.

AZ31b magnesium which is lighter and stronger than aluminum. These materials came at a higher cost than the last chassis from a material standpoint, but the machining methods resulted in an overall drop in the manufacturing costs.

Using magnesium in the chassis delayed the build by almost a month, as there are not many common suppliers for 1/8" magnesium plate. The time delay did not alter the competition plans since the drawings were sent to Team Whyachi well in advance of any intended competitions. In addition to waterjet cutting, they also handled the bend on the titanium wedges that worked so well in the first build of Apollyon.

The electrical system from the last build of Apollyon was not disassembled, allowing for a very short build time for the latest incarnation. I am still using the same six cell A123 packs from Battlepacks, 18V Dewalt drill motors, and Holmes Hobby BR-XL speed controllers. Keeping the electrical system together meant that the build time between having a pile of chassis components and a drivable machine was less than a day.

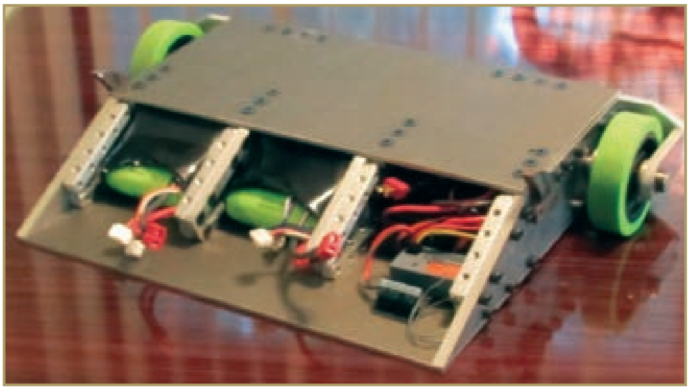
Apollyon does not always compete as a 12 lb robot. The original Apollyon competed at Battlebots in the college 15 lb class, and the newest version was used in a robot hockey competition with a 15 lb weight limit. To take advantage of this room, the gap between the inner supports and outer rails was kept uniform. This allows a second 2,300 mAh battery pack to be installed and used when longer run time or higher current capacity are desired. A special adaptor is used to allow both packs to be plugged into the main power lead in parallel.

In addition to competing in the bot hockey competition, Apollyon competed in the 12 lb class at Robot Battles 42 which took place in Atlanta, GA during Dragon*Con.

Apollyon finished the tournament with a 5-0 record winning the 12 lb class. The floor at Robot Battles consisted of several movable stage pieces placed together to form the combat surface. This unique floor results in large seams and a great deal of flatness issues that would normally render a low wedge like the one used on Apollyon useless. To compensate for the floor, I added large washers to the front bolts that lifted the entire wedge 1/8" off the ground. This combined with the increased traction due to the carpeted surface allowed Apollyon to lift the wedge off the ground under acceleration and drive smoothly over the floor seams.

Video from Robot Battles, as well as many other events can be found on my YouTube page which is located at **www.youtube.com/user/mikencr**. **SV**

Optional battery configuration when competing in classes with a weight limit higher than 12 lbs.



Apollyon after Dragon*Con 2011 with the wedge lifters still attached.

PARTS IS PARTS:

Fine-Tuning a Brushless Electronic Speed Controller

● by Pete Smith

My 12 lb Hobbyweight bar spinner Surgical Strike had a problem at the last couple of events. If I tried to spin up quickly, the blade would seem to "stall" at about half speed and stay there unless I backed the throttle off and then backed up again to full speed. It would then run up to its full speed.

In most fights, this wasn't really a problem since there was time to get it spun back up between hits. However, if the opponent was a good driver they could be attacking when the blade was stalled and perhaps even stop the blade completely, getting me jammed against the arena bumpers. Surgical Strike's only defense is its spinning blade and if an opponent could get — and keep — it stopped, they would usually win.


FIGURE 1

I had noticed this problem first at Franklin in 2010 and then again this year at Motorama. I was competing with several other bots, so never got around to working out what the problems were at the events. I decided to track it down over the summer and if possible, get it fixed before this year's event at the Franklin Institute.


FIGURE 2

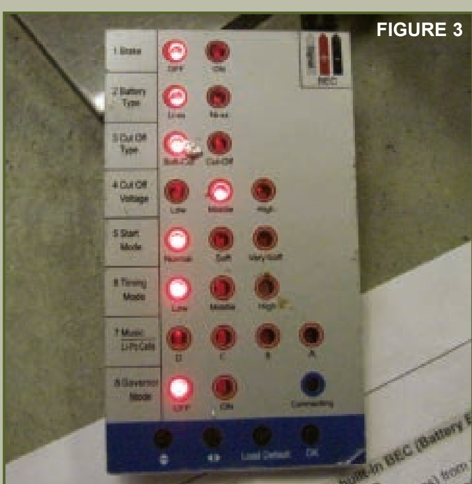


FIGURE 3

I had three main suspects. The first and second were purely mechanical. First, a loose pulley on the motor or the drive shaft could be slipping under the heavy load at start-up but then stop slipping as the blade got up to speed and the load decreased. The second possibility was the motor's bell housing slipping on the motor's shaft. This is a known problem and can usually be fixed by tightening and using Loctite on the screw that secures the bellhousing to the shaft.

The third possibility was that the settings of the Electronic Speed Controller (ESC) were incorrect. I had used a variety of controllers over the last few years but had mostly used the Turnigy Plush 80A. It had worked with few issues. The speed controllers are programmed using a simple programming card, and you can adjust a variety of settings. I suspected that perhaps I was not using the right settings and it was this that was causing the stalling problem.

The ESCs we use are designed for use in model aircraft or cars, so there are no recommended settings from the manufacturer for quickly and smoothly spinning up heavy metal blades. I decided to methodically try all the options and see which worked best. First, I needed a safe setup so that I could spin the blade up without risking an accident if a tooth or even the whole blade came loose.

I used three sections of my Antweight arena with its 3/8" Polycarbonate (PC) walls as an enclosure, and used another old section of PC and some zip ties to secure the bot in a stationary position (**Figure 1**). I screwed a square of thin UHMW under the weapon's axle to reduce friction and to stop the axle from wearing a hole in the PC sheet.

To allow me to check for mechanical slippage, I used a black sharpie to draw lines (**Figure 2**) across the joints of the various

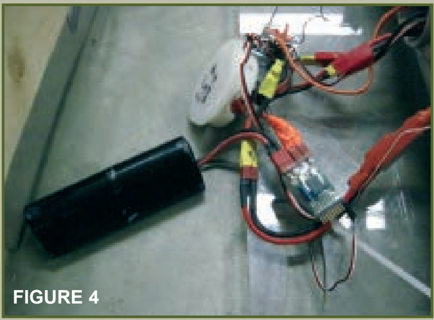


FIGURE 4

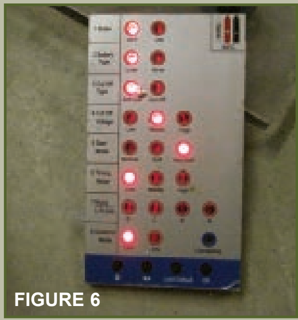


FIGURE 6

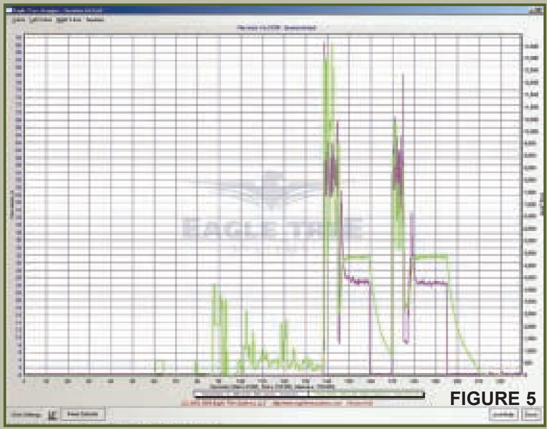


FIGURE 5


FIGURE 9

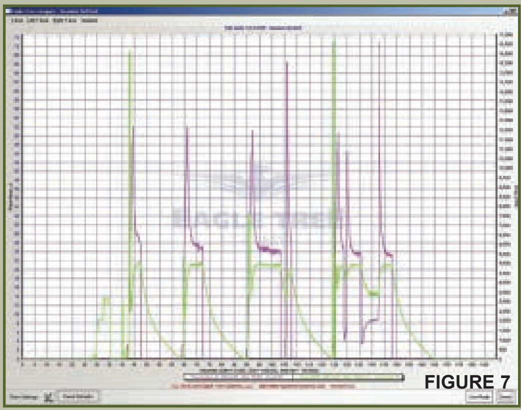


FIGURE 7

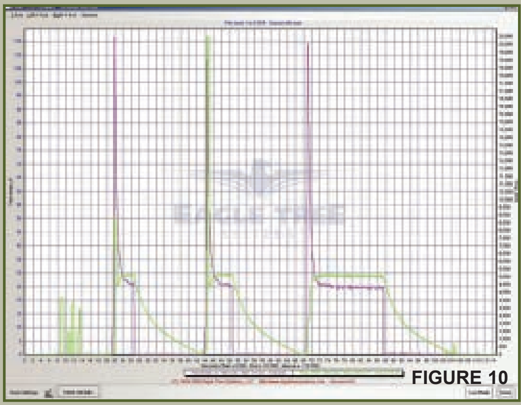


FIGURE 10

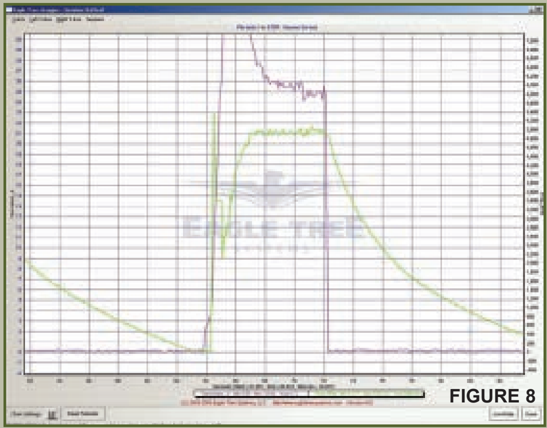


FIGURE 8

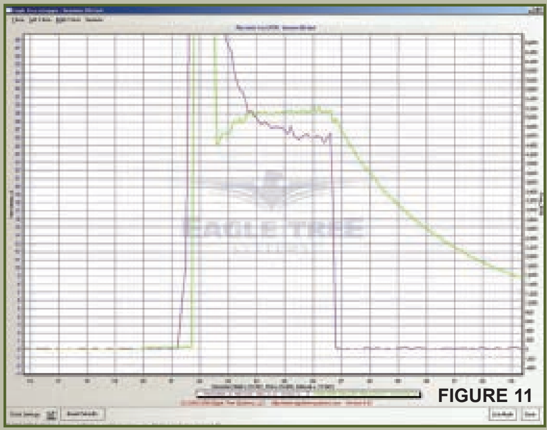


FIGURE 11

components. If the parts moved relative to each other, the lines would be broken and the problem area revealed. A couple of test runs displayed the same problem powering up, but when all the lines were checked there was no slippage to be found.

Now that I had established there was no mechanical problem, I looked at the settings in the ESC. I plugged in the optional programming card (I consider these a "must have" as they make life so much easier) following the supplied instructions and took a picture (**Figure 3**) so that I had a record of those settings.

I removed the programming card and then attached my E-Logger (**Figure 4**) so that I could record the RPM of the motor and the amperage used. (I described the use of the E-Logger in the June '10 issue of *SERVO*.)

I then did a test run, moving the throttle quickly to maximum and letting it run for a few seconds to get up to full speed, then letting it stop and then repeating the process two or three more times. As expected, the blade stalled at half speed during the initial power-up and had to be throttled back down and back up again to get it to reach full speed.

I then shut the bot down completely and reprogrammed the ESC, changing just one thing at a time for first the "Start Mode" and then the "Timing Mode." I repeated the test until I had run through all nine possible combinations. Each time I took a picture and each time the E-Logger would record the run.

Some combinations would not work at all, others would stall like in the original settings, and some would power-up smoothly taking varying amounts of time.

I downloaded the E-Logger data using the supplied software and cable, and then had a look at the graphs of the various runs. Three were of particular interest. The first

run graph (**Figure 5**) shows that the periods of "stalling" at half speed were, in fact, drawing over 50A and with peaks well over 80A. However, once the blade had settled down, it cruised at about 25A and at about 5,000 RPM.

The third run with the Start Mode set at "very soft" (**Figure 6**) and Timing Mode at "low" gave a smooth and uneventful run up to full speed (**Figure 7**) with peak current of about 70A, and again cruising at about 25A and 5,000 RPM. By taking a closer look at the graph (**Figure 8**), it shows that the blade was taking about four seconds to reach full RPM.

The final run of interest was number five. Start Mode was set at "normal" and Timing Mode at "high" (**Figure 9**). This gave a rapid and smooth run up to full speed, peaks of over 115A, and again cruising at 25A and 5,000 RPM (**Figure 10**). A closer look revealed that the blade was reaching full speed in only two seconds, and 90% of full speed in about one second (**Figure 11**).

A quick blade spin-up time is important in combat and I will try out using the same set-up as in run five for the next competition. The high peak amps may be an issue, as might Newton's Third Law of Motion where a rapid spin-up might make the bot spin in the opposite direction for a short time, which may make it hard to drive during spin-up.

The test process has convinced me that it's worth the time to go through all the likely ESC options to find out which is best for any particular bot, and not just settle for the first one that seems to work well enough. SV

So, You Want to Cut Metal on Your Table Saw?

● by Kevin M. Berry

Like many bot builders, I started out as a woodworker. So, I have a shop full of woodworking tools and, now, a serious bot building jones. Some of my woodworking tools work great on metal, like a hand drill, drill press, vice, and screwdrivers. Of course, I had to buy a drill index with metal bits, since brad points don't do so well on steel. I learned a lot about how much sharper metal is than wood when a bit binds up in a hole, and the workpiece does that spinny plate of death number. So, clamping isn't just for glue-ups anymore.

My real learning came when faced with sheets of metal to cut, and no easy access to a shear, arc jet cutter, waterjet, etc. Having a table saw, I thought "Why not?" Thus begins the journey to this short article.

Safety note: I'm a fanatic about doing things right. Face shield, all guards intact, fences, miter gauges, you name it. You should be also!

First, I tried aluminum which is every new bot builder's friend. I started off cutting 1/8" sheet using my plywood thin-kerf blade. I figured tiny teeth, looks like a hack saw, right? Well, it actually cut pretty good, using a slow feed. Pretty soon, however, the teeth sure clogged up.

I also learned something about the heat transfer properties of aluminum. I've never been one to use gloves while cutting on a table saw. I soon learned, though, that the metal sheet got P.D. hot, P.D. quickly. I also (see "spinny plate of death" reference above) paid a lot of attention to what might happen if the cut got bound up. My saw has anti-kickback grippers, but I really don't trust them for wood — much less for slippery metal.

I used a couple mitigations for this: added fences and standing to the side. I clamped a 2 x 4 to the table on the opposite side of the fence, hoping if the plate "kicked," the two fences — plus the anti-kickback pawls — would guide it safely out the door of the shop. (Since my shop is in a cheapo sheet metal shed, I wanted to have a clear line of sight to potential victims before starting the cut.) As far as standing to the side, I've always had a name for anyone who stands in line with material being cut on a table saw: thrill seeker.

I soon graduated to a carbide tipped blade. While there are specialty blades for cutting non-ferrous metals, I've used a regular, high quality, crosscut carbide tipped blade with great success. The little tips scare me, even though I'm pretty sure I've never heard of one flying off when used properly. Still, I took all precautions. (See guards, face shields, standing to the side comments above). Slow feed, steady pressure, push sticks, and clamped guides have combined for many successful cuts — up to 1/2" thick on 6061. The edges are a little rough, but, hey, aluminum cleans up beautifully.

So, next I went for Big Game. Titanium. I bought a bunch, wore out a package of hacksaw blades, and decided to go for it. I wasn't about to pit a steel blade against titanium, especially since anti-

carbide-tipped-steel-blade defense is why I went to titanium in the first place! So, I decided to try an abrasive type metal cutting blade. The kerf is huge on these (1/4" or more) and the cut — I expected — would be ragged. But what the heck, I thought. I moved the table saw out into the yard — luckily, as it turns out.

The first cut was in 1/16" titanium sheet. It ripped right through it, with lots of pretty white sparks. The edge was slagged and ragged, but a quick pass through with the sheet turned over cleaned that right up; sort of like a grinding wheel.

Next, I took on some thicker stock, close to 1/8" thick. By now, a crowd of kids had gathered. "Hey, everybody, Mr. Berry's making lots of pretty sparks!" After making sure the gang was safely back, away we went. LOTS of pretty sparks. And smoke. Then more smoke.

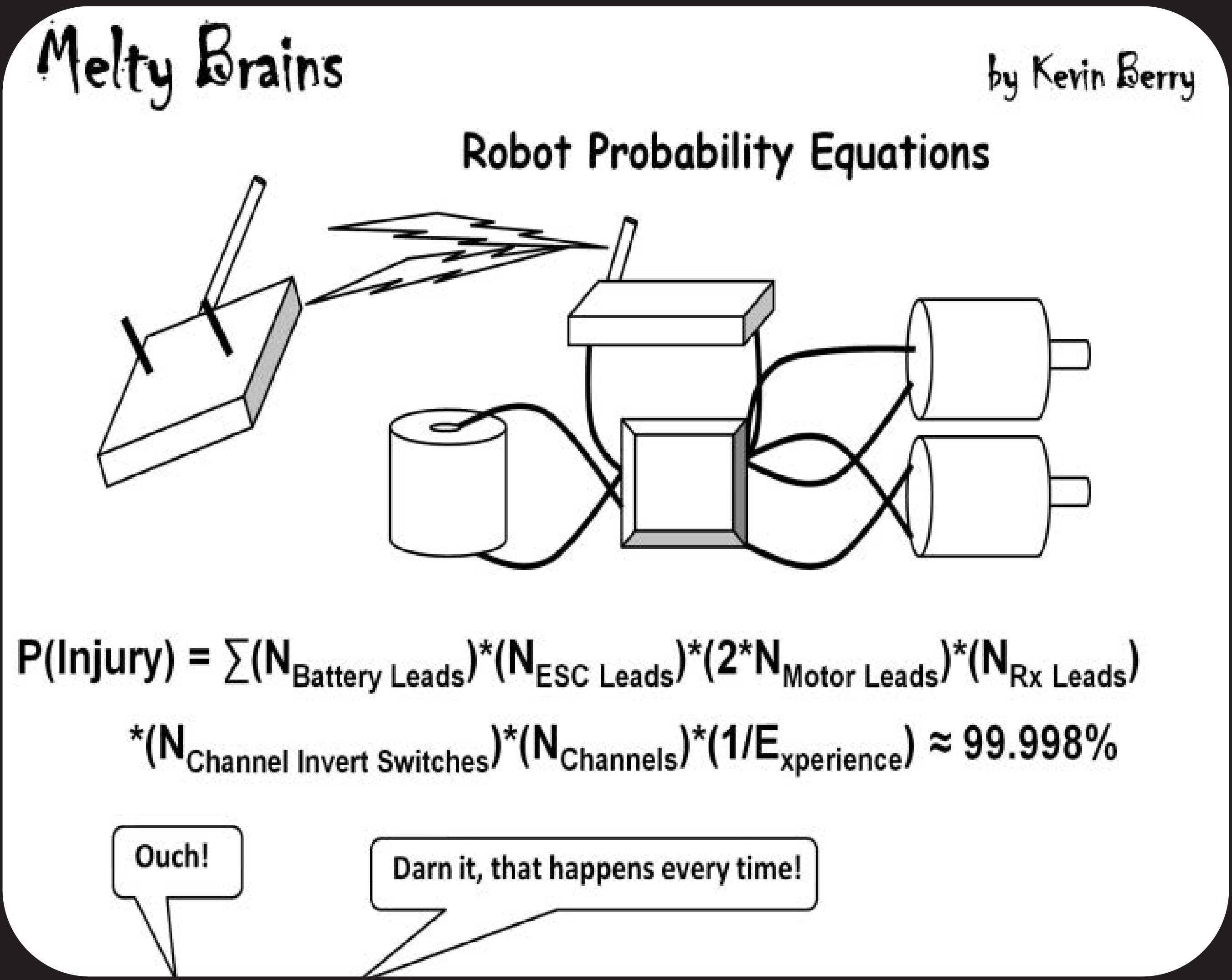
Turns out, gentle readers, you might want to clean the sawdust out of your table saw before making, essentially, thermite. The neighborhood kid pack was treated to a smoldering fire. As a bonus, there was not a hose — nor water tight vessel — anywhere to be found. I did remember to unplug the saw (which was on a GFI outlet that didn't pop) before dumping it on its side and raking out most of the burning wood chips.

So, I now do a complete cleanup before switching from woodwork to metal work. Also, my NEW table saw isn't used to cut titanium. I've discovered the beauty of mail-order waterjet services. **SV**

EVENTS

Completed Events for September 2011

Robot Battles 42 was held at Dragon*Con in Atlanta, GA, on September 4th and 5th. **SV**



An RF Play Day With Sunray

by Fred Eady • www.servomagazine.com/index.php?/magazine/article/november2011_Eady

The very first electronic building block was called a vacuum tube. If you hail from England, that first electronic building block we call a tube is (still) called a valve. The tube — or valve — was followed by the transistor. Compared to the vacuum tube, a typical transistor was smaller, cool to the touch, and could operate without high voltage power supplies. In reality, tubes had already done everything transistors were about to do. Tubes drove the first computers, amplified the first Hi-Fi recordings, and bent the Earth's magnetic field as radio waves. The Beatles recorded their first hit song using a valve-based audio console.

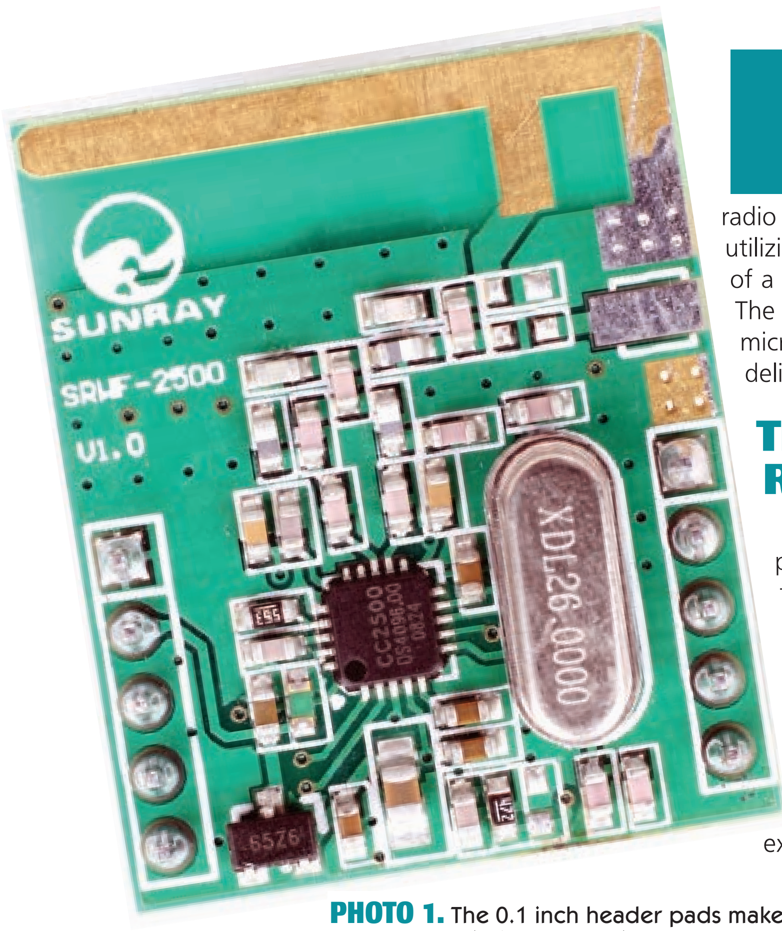
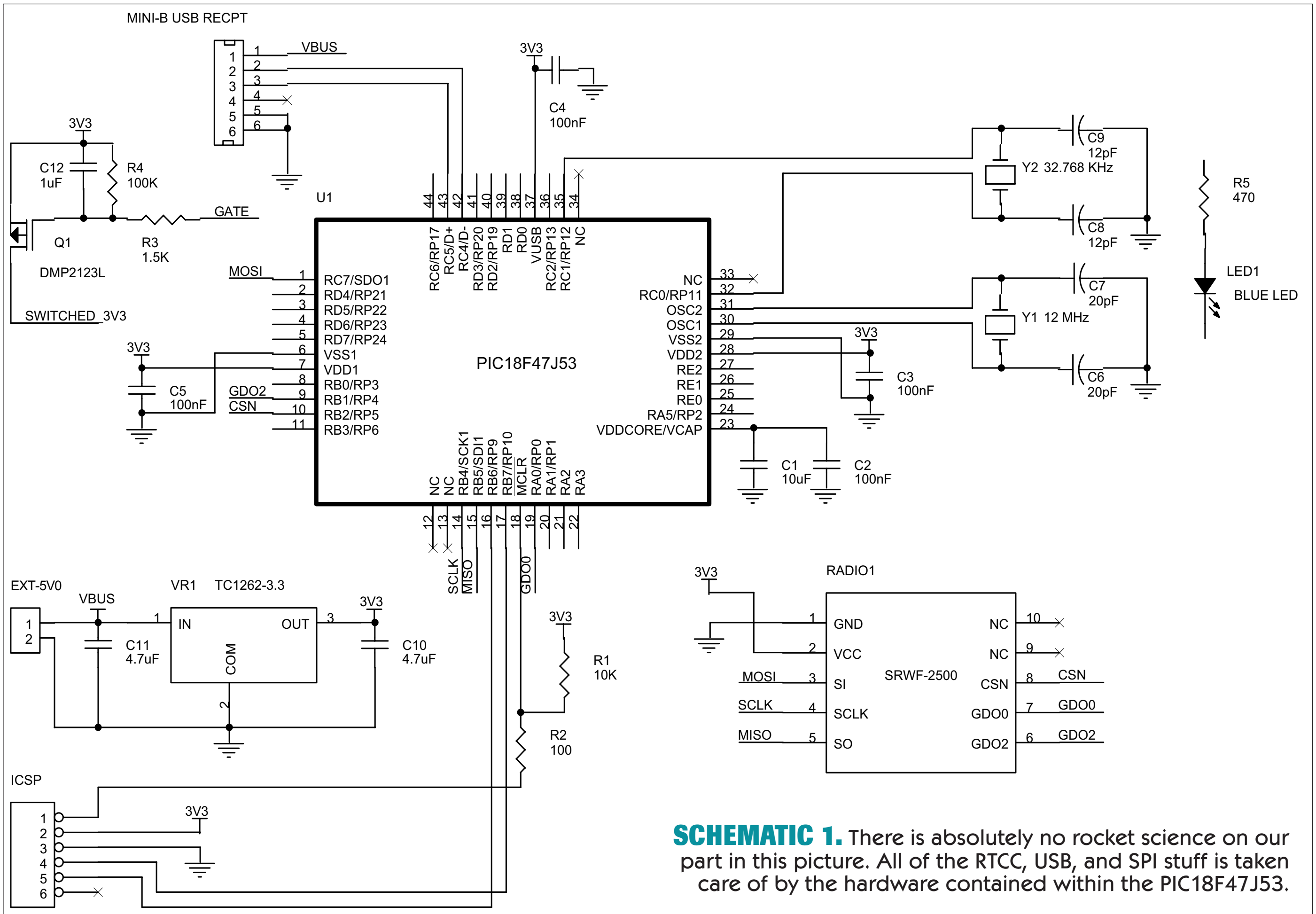


PHOTO 1. The 0.1 inch header pads make the Sunray Technology SRWF-2500 wireless transceiver very easy to adapt to solderless breadboards.

It didn't take long before transistors were ganged into electronic building blocks called integrated circuits. The integrated circuit, or IC, led to solid-state operational amplifiers, elementary logic modules, programmable logic modules, microcontrollers, and radio modules. This month's discussion will be centered on utilizing the transistor building block from the perspective of a low power RF module and a microcontroller or two. The RF building block is based on the TI CC2500 and the microcontroller building block is one of the 10 billion units delivered by the folks at Microchip.

The Primary Sunray RF Building Block

The RF engineers at Sunray Technology have produced a pair of 2.4 GHz radio modules based on the venerable Texas Instruments CC2500 low power radio module. The SRWF-2500 wireless transceiver is Sunray Technology's base 2.4 GHz module. As you can see in **Photo 1**, the SRWF-2500 is a compact RF building block built up on 0.1 inch headers. Aside from any fancy proprietary RF glue circuitry, there is only one way to wrap circuitry around a CC2500 RF IC. In that respect, the SRWF-2500 module is no exception. The good news is that we don't care how



SCHEMATIC 1. There is absolutely no rocket science on our part in this picture. All of the RTCC, USB, and SPI stuff is taken care of by the hardware contained within the PIC18F47J53.

the folks at Sunray designed this wireless transceiver. All we want it to do is allow chatter to flow on an RF link between what we perceive as robotic devices. The SRWF-2500 wireless transceiver was designed to be an RF building block that can easily be incorporated into designs that require a low power radio. Interfacing to the SRWF-2500 is as easy as attaching it to your microcontroller's SPI portal. The wireless transceiver supports a standard four-wire SPI connection (MISO, MOSI, SCLK, and CS). Many of the newer super low power microcontrollers operate at voltages of 3.3 volts and below. The SRWF-2500 can receive and transmit when supplied with a power supply voltage between 3.0 and 3.6 volts. Okay. Now we know that the Sunray wireless transceiver is a bundle of silicon and FR-4 that consists of a 2.4 GHz low power RF IC, an integrated printed circuit board antenna, and an SPI interface. Let's hook it up.

SRWF-2500 Wireless Transceiver Supporting Hardware

What's missing in **Schematic 1**? While you're pondering that question, let's do a slow drive around the circuit. Regardless of the extent of trickiness in a circuit design, one still has to provide a clean and suitable power source. Odds are that your mobile robotic creation will source the circuit depicted in **Schematic 1** with battery power. However, I really don't know what you're going to do. So, the option of powering the radio module from your PC's USB portal is a design point of the circuit you're following in **Schematic 1**. Tying the wireless transceiver's SPI portal to the PIC18F47J53 is the only peripheral connection in our support design. Note that the MASTER SPI SDO/MOSI and SDI/MISO pins are crossed at the SRWF-2500 wireless transceiver, just as the TX and RX pins would be in an RS-232 implementation.

The PIC18F47J53 houses a silicon RTCC (Real Time Clock Calendar). We can put it to use since we have mounted a 32.768 kHz crystal to drive the RTCC clock circuitry. A 12 MHz clock crystal is an ideal choice for the PIC18F47J53 because we can derive a maximum 48 MHz CPU clock, and at the same time provide the necessary clocking to drive the PIC18F47J53's USB engine.

I assembled the hardware in **Photo 2** using an EDTP PIC18F47J53 prototype board mounted on a 0.1 inch pitch EDTP universal breadboard. As you can see, the wireless transceiver fits right in and mounts handily on the

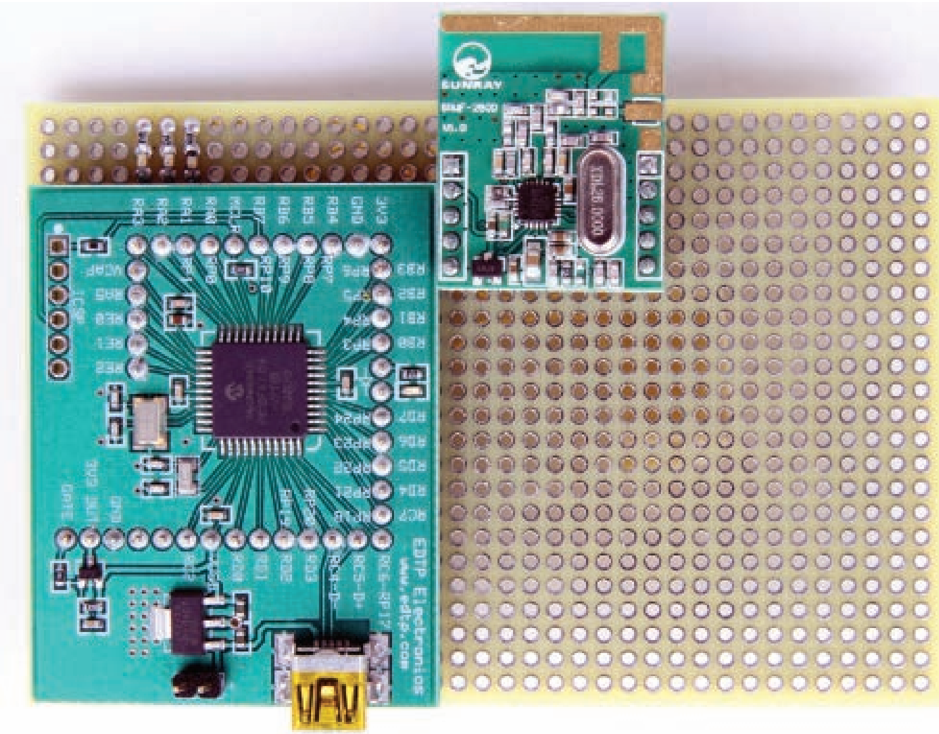


PHOTO 2. In less than an hour, I had this PIC18F47J53 controlling the Sunray Technology SRWF-2500 wireless transceiver with ASCII feedback provided by the PIC's USB portal.

breadboard. Everything is connected using wirewrap techniques. Just in case you haven't figured out what's missing in **Schematic 1**, it's also missing in **Photo 2**. Here's a hint: MAX232.

A Sunray Technology SRWF-2500 Wireless Transceiver Driver

Being a Chipcon part originally, the early CC2500 drivers were written for the most popular microcontroller on the planet at that time: the 8051. Understandably, the current crop of CC2500 drivers focuses on the TI line of low power microcontrollers. Microchip has their own line of low power radio modules. So, why should the Microchip folks write a driver for the CC2500? As I sit writing this, a Woodstock documentary is underway on the television. The idea behind Woodstock was peace and love. With that, let's throw some peanut butter into the chocolate and meld the TI and Microchip points of view.

I managed to tap out a PIC-based Sunray Technology SRWF-2500 driver with CCS C using a mixture of the early Chipcon 8051 and current Texas Instruments MPS430 CC2500 driver code. Although I targeted the PIC18F47J53, the Sunray driver source code can be adapted to just about any other PIC microcontroller.

The probable absence of an Intel or AMD-based PC riding on your mobile monkey is one of the reasons for shelving an RS-232 portal in this design. However, during the debug phase of RF outfitting your robotic device, having the ability to "talk" to the PIC18F47J53 is essential. So, I included access to the PIC's USB portal instead of installing a standard MAX232 serial port. The idea is to use the PIC18F47J53's USB portal to emulate an RS-232 type of communications link in CDC mode.

To complete the RS-232 emulation, the PC will be forced to post a virtual COM port at the end of its USB link via a Windows driver. Basically, we're using the USB features of the PIC18F47J53 to replicate the services provided by an FTDI USB-to-UART device.

With the USB RS-232 emulation hardware in place, all we need to provide is a suitable firmware driver. Fortunately, we don't have to choke up any specialized USB driver code from scratch. Our USB CDC link can

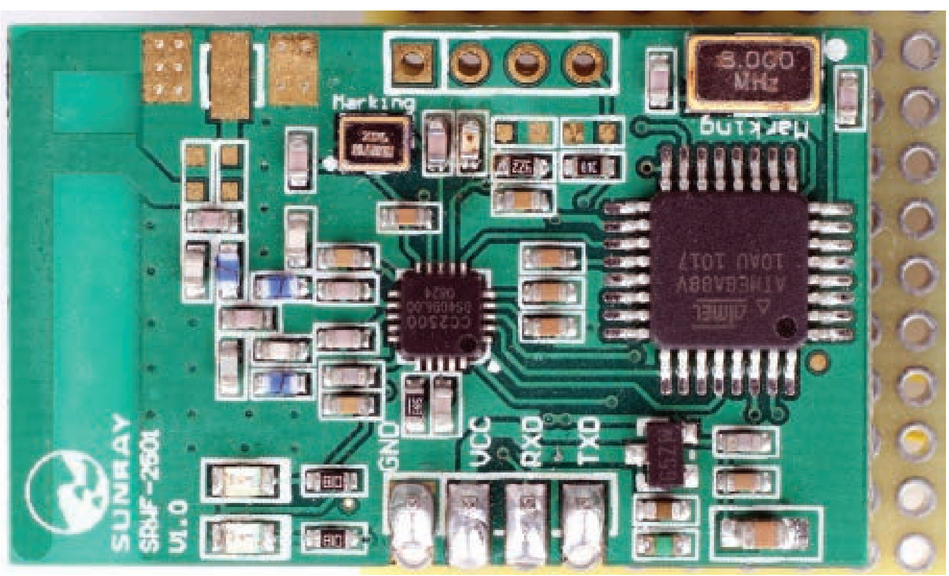
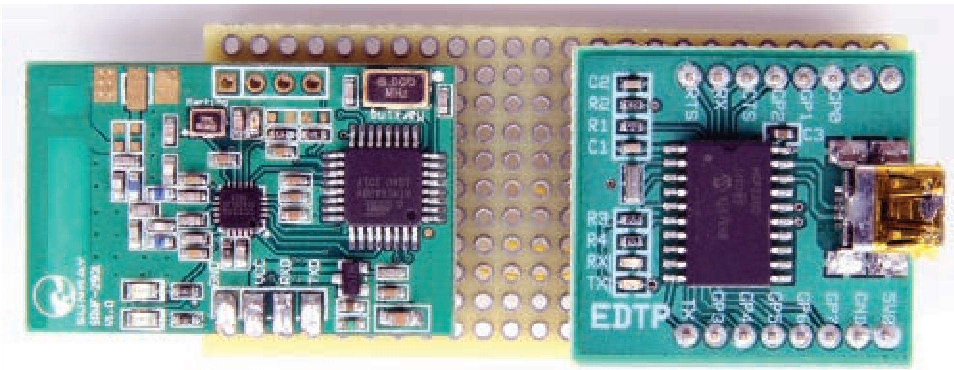


PHOTO 3. The SRWF-2501 wireless transceiver module is an SRWF-2500 wireless transceiver coupled with an on-board Atmel ATMega88 microcontroller filled with a Sunray Technology CC2500 driver.

PHOTO 4. This is a shot of an SRWF-2501 wireless transceiver module with a Microchip MCP2200 USB 2.0 to UART protocol converter with GPIO handling the data I/O duties.



be established with this simple line of code:

```
#include <usb_cdc.h>
```

From a programmer's standpoint, implementing USB has never been easier. By including *usb_cdc.h* in our wireless transceiver driver, we gain program access to a library that adds a virtual COM port to a PC via the standard USB Communication Device Class (CDC). Everything we need is included with the library, which consists of the USB code, USB descriptors, interrupts, and handlers.

The really cool thing about the CCS USB CDC library is that the library calls have the same syntax as their RS-232 cousins. For instance, the *usb_cdc_kbhit()* function returns a Boolean TRUE if there are received characters waiting to be fetched from the receive buffer. Other CDC library functions that may be familiar include:

```
usb_cdc_getc()
Get a character from the receive buffer
usb_cdc_putc()
Put a character into the transmit buffer
usb_cdc_puts(*ptr)
Send a null terminated string to the transmit
buffer
```

The *usb_cdc_putc()* will wait to execute if the buffer is full. To alleviate the pain of waiting, the CCS CDC library offers the *usb_cdc_putready()* function which returns a Boolean TRUE if there is room in the transmit buffer for another character. On the other hand, if the programmer is impatient and any type of waiting is unacceptable, the *usb_cdc_putc_fast*() tries to stuff a character into the transmit buffer regardless of available space. Using the fast library call invokes *Fred Eady's First Rule of Embedded Computing* which states that "Nothing is free." If no room for a character exists in the transmit buffer upon issuing the *usb_cdc_putc_fast*() call, the forced character is ignored and dumped into the bit bucket.

The Sunray wireless transceiver driver that I created initializes the PIC18F47J53, handles the CDC communications link, implements a receiver function, and implements a transmit function. The transceiver driver's *main()* function contains code that shows you how to use the CDC library calls within a receiver function. A transmitter function can also be selected to run inside of the driver's *main()* function. The transmitter function demonstrates how to build and send a data packet.

The SRWF-2500's CC2500 can be instructed to take a temperature measurement. The wireless transceiver driver includes a skeleton function called *get_temp()* to help you use the CC2500's temperature feature.

Alternate SRWF-2500 Wireless Transceiver Implementations

The SRWF-2501 wireless transceiver module mounted on a piece of universal breadboard in **Photo 3** is capable of operating on eight channels with baud rates of 1,200 bps,

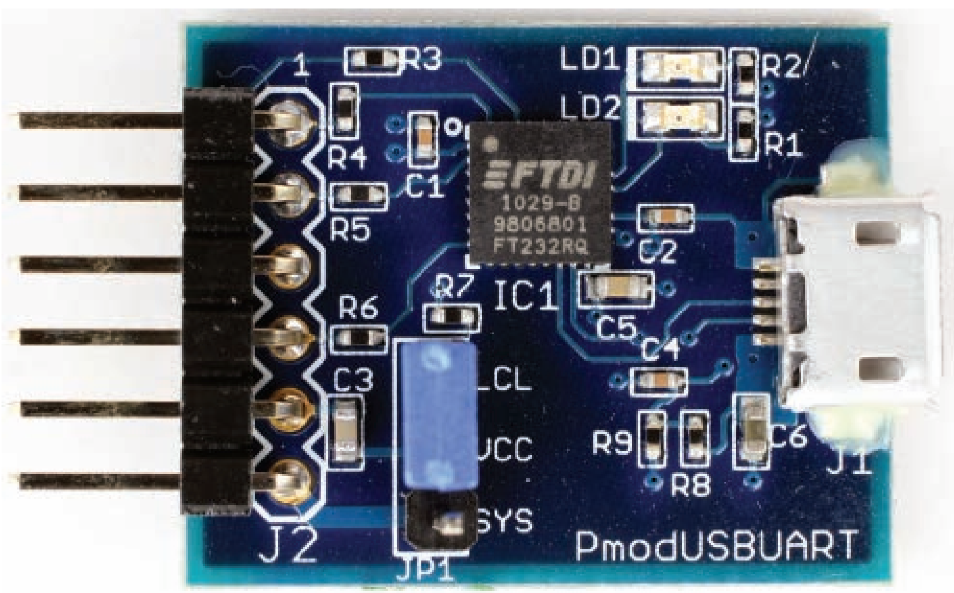


PHOTO 5. If you're not using the FT232RQ because of its difficult-to-solder-in-the-garage QFN package, the Digilent PmodUSBUART will put an end to that excuse. All of the RS-232 signals you need to communicate with the SRWF-2501 wireless transceiver module are available on J2.

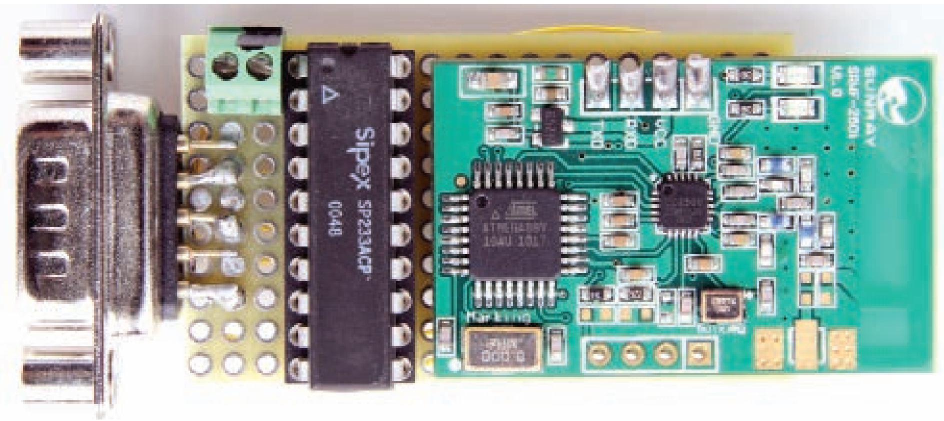


PHOTO 6. That SP233ACP works just like the legacy MAX232 but sans charge pump capacitors. This lashup is perfect for adding RF capability to pre-USB equipment.

2,400 bps, 4,800 bps, 9,600 bps, 19,200 bps and 38,400 bps. The channels and baudrates are software selectable.

The wireless transceiver module is designed to communicate with a host microcontroller's UART using TTL voltage levels. VDD, GND, TX, and RX are the only physical connections the SRWF-2501 module accepts.

On-module intelligence for the SRWF-2500 is provided by an Atmel ATMega88. The ATMega88 is a general-purpose microcontroller endowed with 8 KB of Flash and 1 KB of SRAM. The ATMega88's Flash contains the Sunray Technology version of a CC2500 driver, as well as the baud and channel configuration firmware. The Sunray Technology CC2500 driver is packet oriented and can only belt out a full CC2500 buffer per transmission. So, the Sunray module is perfect for small packet low rate data transfer applications.

It's a no brainer to deduce that a copilot microcontroller can be attached to the transceiver's TTL-compatible serial interface. The ATMega88's firmware provides an aim and shoot platform. That is, every character that enters the module's TX input is transmitted without any intervention by the host. Likewise, any data received by the SRWF-2501 module's CC2500 is automatically presented to the RX output.

In that the Sunray Technology module is "smart," we can deploy the SRWF-2501 in a number of interesting ways. For example, **Photo 4** captures an SRWF-2501 module tied to a Microchip MCP2200 USB 2.0 to UART protocol converter with GPIO. This particular configuration allows the Sunray Technology module to attach to a PC's USB portal.

When using the SRWF-2501 wireless transceiver module with a PC, the Sunray Technology folks recommend using a free terminal emulator program called AccessPort. AccessPort is pretty handy as it can send and receive data in hex and ASCII modes. The AccessPort terminal emulator is perfect for sending the short command strings that configure the SRWF-2501 module's baud rate and channel selection.

Look up "handy" in the dictionary and you'll find that its definition is Digilent Pmods. If you've worked with the Digilent Cerebot 32MX7, you may have a PmodUSBUART lying around on your bench. The PmodUSBUART is based on the FTDI FT232RQ USB-to-UART bridge and is the replacement for the PmodRS232. The local (LCL) and system (SYS) jumper points in **Photo 5** allow you to control the power associated with the PmodUSBUART. Basically, if the equipment that is attached to the PmodUSBUART is powered on its own, the jumper should be set to LCL. Otherwise, jumpering to SYS allows the PmodUSBUART to supply +3.3 volts to the attached equipment on its SYS3V3 pin. You can get all of the scoop on the PmodUSBUART from the Digilent website.

It took quite a while for me to embrace USB. There may be some of you out there that are still on the fence. Or, maybe your piece of robotic equipment does not have USB capability. You can wire up a MAX232 and a sextet of charge pump/bypass capacitors. Or, you can drop a Sipex SP233ACP in its place. The SP233ACP doesn't require any charge pump capacitors. My SP233ACP-equipped SRWF-2501 wireless transceiver module is the subject of **Photo 6**.

You Can Have It All

Behold **Photo 7**. It's a SRWF-2500 test board complete with a Silicon Laboratories CP2102-based USB portal, a baud rate/channel

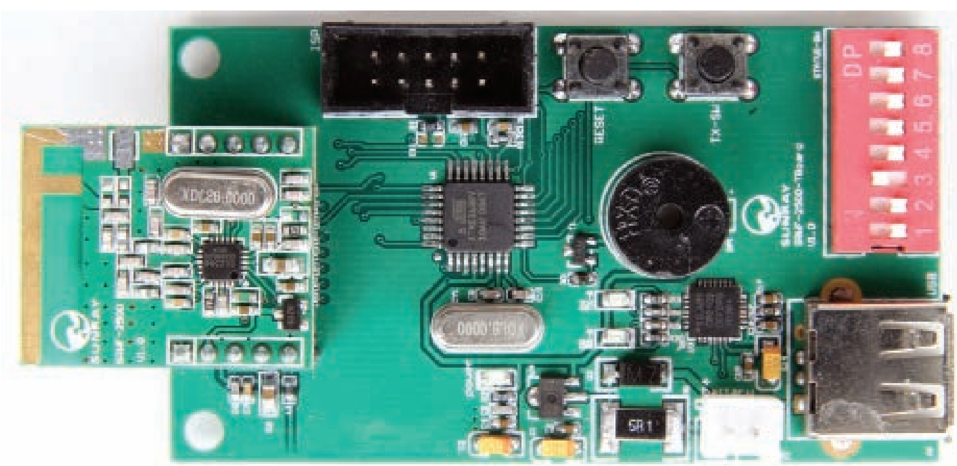


PHOTO 7. For those that want it all and want it right now!

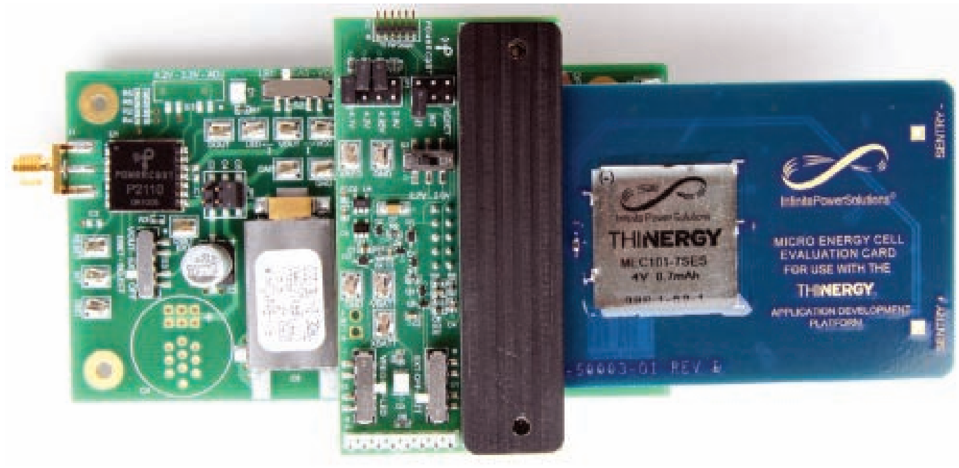


PHOTO 8. Powering RF with RF. That's like making chickens with chicken eggs. What came first, the receiver or the transmitter??

DIP switch, a piezo speaker, an ATMega88 filled with CC2500 driver/test/configuration code, and the SRWF-2500 wireless transceiver. If you're feeling bully, the test board provides ISP programming portal access for the ATMega88. Danger, Will Robinson!!

Used in conjunction with AccessPort, the prospective SRWF-2500 wireless transceiver user can establish a communications link between a pair of SRWF-2500 test boards. Pressing the TX-SW pushbutton sends a canned message to the receiving test board which kicks off a squawk of ASCII data from the receiving node. Thus, the test board can be used for range testing utilizing various baud rates and RF channels.

Powering RF Using RF

What you see in **Photo 8** is a battery charger that derives its power from RF. The pA dipole or patch antenna is attached to J1. Encoded 915 MHz RF energy is emitted by a local 915 MHz Powercaster transmitter and is fed to the Powercast P2110 Powerharvester receiver by the antenna servicing J1. The Powerharvester receiver converts the received RF energy and stores it into the large holding capacitor. The power converted and stored by the receiver is used to charge the THINERGY energy cell.

So what, you say? Well, what does the **Photo 8** P2 header legend say? It say's *TI RF2500*. That 2500 is a reference to the CC2500. Thus, this Powercast circuitry and THINERGY energy cell has the ability to power a CC2500-based radio platform.

Good Day, Sunray

That's what the song lyrics would have been if John, Paul, George, and Ringo had to implement a low power 2.4 GHz communications link. Being a *SERVO* columnist, it's my job to expose you to every bit of robotic-possible electronic gadgetry that I can. This month, you've taken a tour of the RF technology offered by the folks at Sunray Technology. I'll make my CC2500 driver firmware available via the *SERVO* ftp site. **SV**

Fred Eady can be reached via email at fred@edtp.com.

Sources

Lemos International
SRWF-2500 Wireless Transceiver
SRWF-2501 Wireless Transceiver Module
SRWF-2500 Test Board
www.lemosint.com

CCS, Inc.
CCS C Compiler
www.ccsinfo.com

Microchip
PIC18F47J53
MCP2200
www.microchip.com

Powercast
Powercast P2110 Powerharvester Receiver
www.powercastco.com

Designing A Low Cost Laser – Part 2

by Joe Grand

www.servomagazine.com/index.php?/magazine/article/november2011_Grand

Last time, I introduced the Laser Range Finder (LRF) project and discussed my initial work in designing a low cost module that can detect the distance to a target object using a laser and camera. I had proved the concept of using optical triangulation and was successful in obtaining an image with my prototype circuitry. The article continues here with a look into the final hardware design and functionality of the device.

LRF Hardware

Confident that the camera interface was functioning properly, it was time to finalize the design and move away from my Propeller proto board-based development platform onto a custom PCB containing only the electronics required for the LRF (**see Figures 1A** and **1B**).

The PCB for the LRF module is four layers and measures 3.95" x 1.55" (~100 mm x 39 mm). All components are mounted on the back side of the board with the exception of the camera module (which is so tiny it can barely be seen on the left side of the PCB) and the laser diode. The Propeller, its supporting electronics, and the camera interface circuitry can be seen on the right of the back side image; the laser diode control circuitry is on the left. A Prop Clip/Plug four-pin interface (used for Propeller programming) is at the top of the board as surface-mount pads. Below it is a bi-color (red/green) LED for indication of system status. At the bottom is a single row, four-pin header (serial IN, serial OUT, VCC, GND) for the primary user interface (sending commands and receiving data).

The center points of the camera and laser diode are spaced at exactly 78 mm which closely matches the spacing of my proof-of-concept (and the original project on which *my* project is based), and keeps the module to a reasonable size. The spacing of the camera and diode ultimately determines the effective range (minimum and maximum detectable distance).

The major hardware components include the Propeller, serial EEPROM for Propeller firmware and unit configuration data, OVM7690 camera, PCA9306 level translator, laser

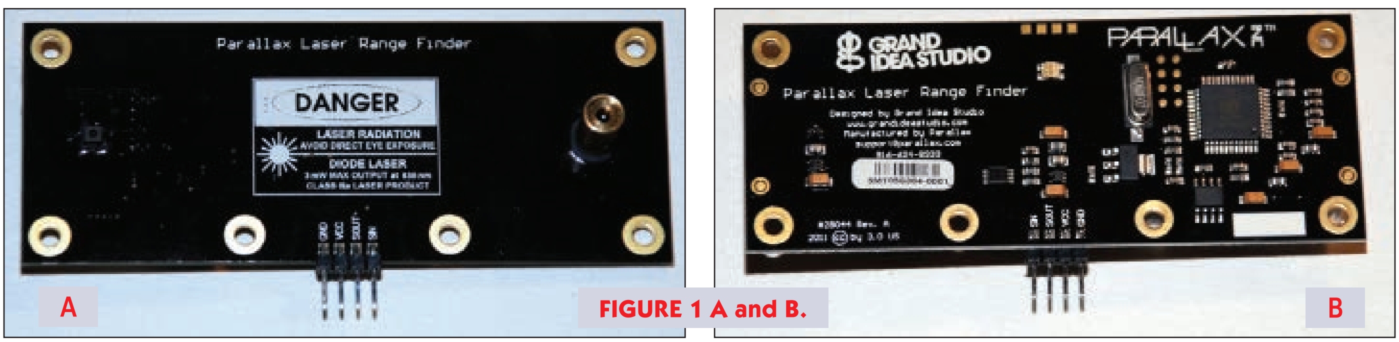


FIGURE 1 A and B.

Range Finder

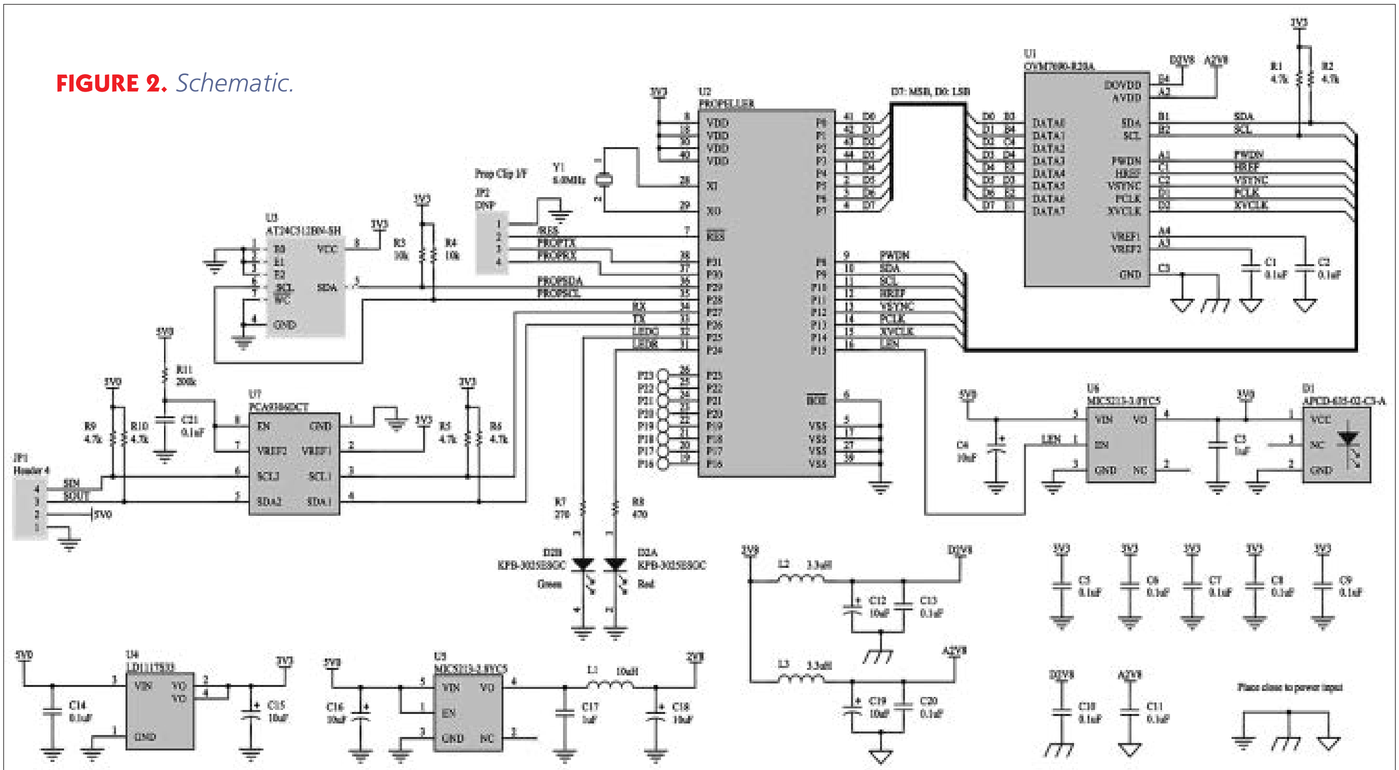


FIGURE 2. *Schematic.*

diode, and three LDO linear regulators. Refer to the **schematic** and **Bill of Materials** for complete details.

LRF Commands

The LRF is controlled through a simple serial communications interface (ASCII command/responses at TTL-level). Host baud rate is automatically detected upon LRF power-up when the user sends a "U" character, allowing easy interfacing to a variety of microcontrollers and computers. The auto-detection routines are a slightly modified version of Raymond "Rayman" Allen's RS-232 interface for uOLED-96-Prop (**www.rayslogic.com/propeller/3rdPartyHardware/uOLED-96-Prop/RS232Driver.htm**) and support standard baud rates from 300 to 115.2K.

The LRF currently has 12 commands:

Basic Commands

- **R** Single range measurement (returns a four-digit decimal value in millimeters).
- **B** Single range measurement (returns a four-byte binary value in millimeters).
- **L** Repeated range measurement (any subsequent byte will stop the loop).
- **E** Adjust camera for current lighting conditions.
- **S** Reset camera to initial settings.
- **V** Print version information.
- **H** Print list of available commands.

Qty	Reference	Manufacturer	Manuf. Part #	Distributor	Distrib. Part #	Description
13	C1, C2, C5, C6, C7, C8, C9, C10, C11, C13, C14, C20, C21	Kemet	C0603C104K4RACTU	Digi-Key	399-1096-2-ND	0.1 µF bypass capacitor, 16V, X7R, 0603
2	C3, C17	Kemet	C0805C105K8RACTU	Digi-Key	399-1172-1-ND	1 µF ceramic capacitor, X7R, 10V, 0805
6	C4, C12, C15, C16, C18, C19	Kemet	T491A106M016AS	Digi-Key	399-3687-1-ND	10 µF capacitor, 20%, 16V, tantalum, size A
1	D1	Arima Lasers	APCD-635-02-C3-A	Ayase America	N/A	Laser diode module w/APC, 635 nm, < 3 mW, 6.2 mm D
1	D2	Kingbright	APB3025ESGC-F01	Mouser	604-APB3025ESGC-F01	Red/green bi-color LED, 1201W SMT
1	JP1	Molex	22-28-8040	Digi-Key	WM6004-ND	Single row, right-angle vertical header, four-pin, 0.1” P
1	L1	TDK	MLZ2012M100W	Digi-Key	445-6396-1-ND	Inductor, power, 10 µH, 470 mR, 150 mA, 0805
2	L2, L3	TDK	MLZ2012A3R3W	Digi-Key	445-6394-1-ND	Inductor, Power, 3.3 µH, 340 mR, 200 mA, 0805
6	R1, R2, R5, R6, R9, R10	Any	Any	Digi-Key	P4.7KGCT-ND	4.7K, 5%, 1/10W, 0603
2	R3, R4	Any	Any	Digi-Key	P10KGCT-ND	10K, 5%, 1/10W, 0603
1	R7	Any	Any	Digi-Key	P270GCT-ND	270 ohms, 5%, 1/10W, 0603
1	R8	Any	Any	Digi-Key	P470GCT-ND	470 ohms, 5%, 1/10W, 0603
1	R11	Any	Any	Digi-Key	P200KGCT-ND	200K, 5%, 1/10W, 0603
1	U1	Omnivision	OVM7690-R20A	NuHorizons	N/A	CMOS VGA (640x480) CameraCube, 1/13” color
1	U2	Parallax	P8X32A-Q44	N/A	N/A	Microcontroller, Propeller, TQFP-44
1	U3	Atmel	AT24C512BN-SH-T	Digi-Key	AT24C512BN-SH-TCT-ND	Memory, serial EEPROM, 64 KB, SOIC-8
1	U4	ST Microelectronics	LD1117S33	Digi-Key	497-1241-1-ND	Linear regulator, LDO, 3.3V, 800 mA, SOT223
1	U5	Micrel	MIC5213-2.8YC5	Digi-Key	576-2751-1-ND	Linear regulator, LDO, 2.8V, 80 mA, SC70-5
1	U6	Micrel	MIC5213-3.0YC5	Digi-Key	576-2752-1-ND	Linear regulator, LDO, 3.0V, 80 mA, SC70-5
1	U7	Texas Instruments	PCA9306DCTR	Digi-Key	296-18509-2-ND	Voltage-level translator, SM8
1	Y1	Abracon	ABLS-6.000MHZ-B4-T	Digi-Key	535-10208-2-ND	Crystal, 6 MHz, 18 pF, HC49/US SMD
1	PCB	Sonic Manufacturing	LRF A	N/A	N/A	PCB, fabrication, assembly, and test

FIGURE 3. *Bill of Materials.*

Advanced Commands

- **O** Display coordinate, mass, and centroid information for all detected blobs.
- **X** Calibrate camera system for range finding (requires user interaction).
- **G** Capture and send single frame (eight bits/pixel grayscale at 160x128).
- **C** Capture and send single frame (16 bits/pixel YUV422 color at 640x16) with laser enabled.
- **P** Capture and send processed frame (16 bits/pixel YUV422 color at 640x16) with background subtraction.

LRF Image Viewer

While waiting for the prototype PCBs to come back from fabrication and assembly, I started working on a PC-based monitor program called the *LRF Image Viewer* that would give me an easy-to-use graphical interface to control the LRF. Though not a requirement for usage or operation of the LRF (since the module can be controlled with any host microcontroller through its serial interface and can calculate the necessary range finding results onboard), I thought it would be handy for users to have a simple tool at their disposal. The LRF Image Viewer is based on the open source CMUcam3 Frame Grabber tool (**www.cmucam.org/wiki/CMUcam3-Frame-Grabber**) that was originally written in Visual Basic. I ported the tool to Visual Basic .NET and heavily customized it to meet my needs. The primary features include:

- Read system/debug messages sent from the LRF.
- Send commands to the LRF.
- Grab/display/save images (grayscale or color, bitmap or raw binary format).
- Enable PC-side image processing functionality (blob detection and identification, range/distance calculations).

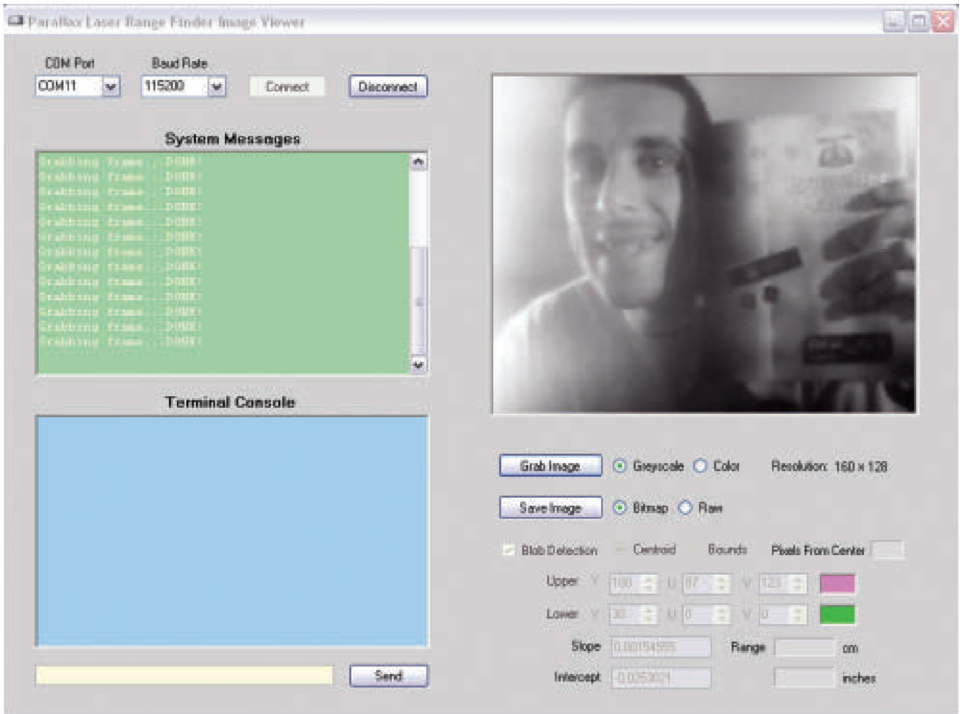


FIGURE 4. *Screenshot of the LRF Image Viewer application running on a PC. I can easily control the LRF using this program and even take frame grabs to see what the camera is seeing.*

Image Processing and Blob Detection

A key function of the LRF is to capture an image with the camera and determine the location of the laser spot (blob) within the frame. Once the blob is detected, its centroid (center of mass) and resulting *pfc* value are calculated and used in the function that determines the distance from the sensor to the target object.

My first attempt at blob detection was based on the CMUCam3's cc3_color_track.c source file, revision 556 from 12/29/2008 (**www.cmucam.org/wiki/Downloads**). This simple function worked well in scenarios where there was a singular blob of pixels within the frame. However, if there were pixels outside of the primary blob — but still within our frame — they were included in the blob's bounding box (which is either a virtual or visual box surrounding the detected blob).

Here's an example of a frame with multiple blobs (the correct laser spot is on the left; a reflection is on the right) and the resulting bounding box:

When the bounding box was calculated (shown as a red border), it started from the left-most pixel that met the defined brightness threshold and ended at the right-most pixel that met the defined brightness threshold. The routine didn't take into account any space in between pixels that didn't match the threshold, as it basically looked at the entire frame and created a box around the largest area of matched pixels.

This caused the bounding box to be much larger than the primary blob (the blob that is most likely to be the laser spot used in the range finding calculations). As such, the centroid was also miscalculated since the routine defined the centroid as the center location of the bounding box and not the primary blob's true center of mass.

I mentioned my problem to Zoz, a friend and former brother in arms of Discovery Channel's *Prototype This* television show. Zoz specializes in human-robot interaction and computer vision — among other things — and he helped devise a plan for a simple, efficient, and robust algorithm that could execute on a dedicated microcontroller like the Propeller, and detect multiple blobs within a frame.

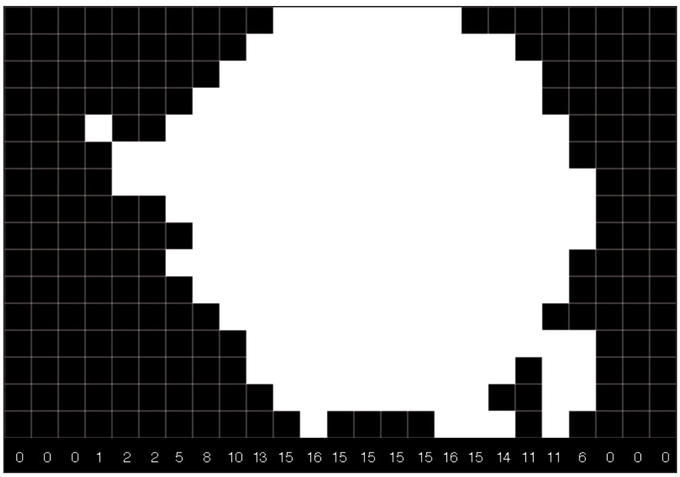
The image processing and blob detection routines function as follows:

1. Background Subtraction. This step occurs during the frame grabbing process. Two consecutive frames are grabbed: one with the laser diode off and one with the laser diode on. Each pixel's Y/luma component from the first frame is subtracted from the same pixel's Y/luma component from the second frame (and absolute valued), leaving only the pixels that have changed in brightness between the two frames. All other background details (anything that has stayed the same between the two frames) disappear. Due to timing constraints of the frame grabber running on the Propeller, the U/V color components are grabbed only on the first of the two frames and not modified. Details of pixel/background subtraction can be found at **http://homepages.inf.ed.ac.uk/rbf/HIPR2/pixsub.htm**.

2. Thresholding. Look at each pixel within the frame and determine if it is above the defined brightness

threshold (currently, any pixel with a Y value greater than 0.3). If so, the pixel is set to '1' (white). If not, the pixel is set to '0' (black).

3. Column Sum. Count the number of 1 pixels within each vertical column. This results in a one-dimensional array containing the number of valid pixels per column. By summing the valid pixels, we can more easily and quickly search to locate any blobs within the frame. The following image shows the zoomed-in blob with the column's sum printed at the bottom of each column:



4. Blob Detection. Traverse the one-dimensional array of column sums looking for any sums above a defined threshold (currently, a column sum needs to be greater than two in order to be considered part of the blob). For example, in the previous image, the blob would start at column 7 (which has a sum of five) and end at column 22 (which has a sum of six). This is repeated across the entire frame until all blobs have been detected.

5. Mass/Centroid Calculation. Calculate the total mass and centroid for the detected blob(s) in the frame. The *mass* is simply the number of valid 1 pixels within the total blob. The *centroid* of a blob is calculated by weighting every valid pixel with where it is in the blob and averaging by the total mass:

For column 1..n of the blob:

sum = 1 * s1 + (2 * s2) + ... + (n * sn)

where sn = column sum for column n

Then, centroid = sum / mass

Simpler image processing — like the CMUcam3's blob detection — sets the centroid as the center point of the bounding box. However, this type of calculation would only return a proper value for round, balanced spots. Since we don't know what type of object the laser will be pointing at and how the light will vary when it reflects off of the object, the resulting blob may neither be round nor balanced. Performing the weighted average gives a more accurate center of mass result, regardless of blob shape.

Here's an example of determining the centroid using the blob from the previous image:

sum = (1 * 5) + (2 * 8) + (3 * 10) + ... + (15 * 11) + (16 * 6) = 1737

centroid = sum / mass = 1737 / 200 = ~8.7

The blob with the largest mass is then chosen as the primary blob (which we assume is the actual laser spot) and will be used for the subsequent range finding calculations. If there are multiple blobs with the same mass, the first occurrence remains the primary.

Calibration

To account for manufacturing and assembly variances — particularly related to the camera and laser diode alignments — each LRF module must go through a calibration step during production. The unit can also be re-calibrated by the user at a later date if desired.

The calibration routine requires the user to place the LRF module at a number of fixed distances (currently 20 cm to 70 cm at 10 cm intervals). The LRF takes measurements at each distance and calculates the SLOPE, INTERCEPT, and PFC_MIN values (the routine is based lightly on **www.eng.umd.edu/~nsw/ench250/slope.htm**). The values are then stored in an unused portion of the non-volatile boot serial EEPROM. The LRF has a 64 KB EEPROM which leaves 32 KB available for data storage after the Propeller uses the first 32 KB for program code. This also means that the values will not get overwritten when the LRF code is re-loaded into the EEPROM.

The SLOPE and INTERCEPT are used to convert the pixel offset to angle using a best-fit slope-intercept linear equation (discussed earlier in this article). The PFC_MIN value is used to set the maximum allowable distance of the LRF module which is represented by a minimum of pixels from center value. Check out this video demonstrating the calibration process:

www.youtube.com/watch?v=1gk_tRbJO84.

FIGURE 5. *LRF measurements.*

Actual Distance to Target (cm)	Calculated Distance (cm)	Difference (Δ)	% Error
20	19.9	0.1	-0.50
30	29.7	0.3	-1.00
40	40.1	-0.1	0.25
50	50.3	-0.3	0.60
60	60.2	-0.2	0.33
70	70.8	-0.8	1.14
			Average % Error
			0.64

Actual Distance to Target (in)	Calculated Distance (in)	Difference (Δ)	% Error
10	9.9	0.1	-1.00
20	20.1	-0.1	0.50
30	30.7	-0.7	2.33
40	40.3	-0.3	0.75
50	48.8	1.2	-2.40
75	70.3	4.7	-6.27
			Average % Error
			2.21

Measurement Results

With the LRF design completed and calibrated, it was time to take a series of measurements to determine how accurate the LRF was and if it was worthy enough to become a Parallax product. I set up 12 marked distances on the floor of my lab and took measurements with the LRF from each location (see **Figure 5**).

In summary, the LRF has a usable range from about six inches to four feet (48 inches). Within the usable range, the change in pixels from center/angle is very noticeable, making the range calculations more reliable. As the distance increases — although the camera can still "see" the laser spot — the change in pixels from center/angle is much harder to determine which causes a reduction in accuracy. The LRF firmware intentionally limits the maximum distance to 100 inches. At distances less than six inches, the laser spot is out of the camera's field-of-view, so no blob detection can occur.

Accuracy within the usable range varies from perfect (no difference between actual distance and the distance calculated by the LRF) to around ±2% error (approximately 1/4" to 1" difference between actual and calculated distance).

Limitations

The LRF is able to consistently and reliably detect multiple blobs and make a determination of which is the primary laser spot. The range finding math works well with reasonable accuracy. However, like any sensor system, it's not suitable for use in all conditions and has its limitations:

- **Range.** As discussed previously, the design is most accurate within its usable range of six inches to four feet (48 inches). Longer distances will result in a noticeable reduction in accuracy, though could still be used for gross measurements or simple object detection.

- **Environment.** The camera system has automatic white balance and automatic exposure enabled by default (they can be enabled/disabled with a single command sent to the LRF module), and will automatically adjust its image to account for sudden changes in lighting conditions. However, the module works best in a controlled environment, such as indoors with minimal changes in brightness across the frame. The module is also less reliable when the laser is shining onto a bright object (for example, a white piece of paper) since the background subtraction done during image processing could potentially subtract the bright laser from the already bright frame. Giving the camera time for its automatic white balance and automatic exposure to settle helps a bit to make the laser spot stand out. Using a red filter over the camera will also help the red laser spot become more visible to the camera in certain situations.

Demonstrations

To show how the LRF can be used in real world

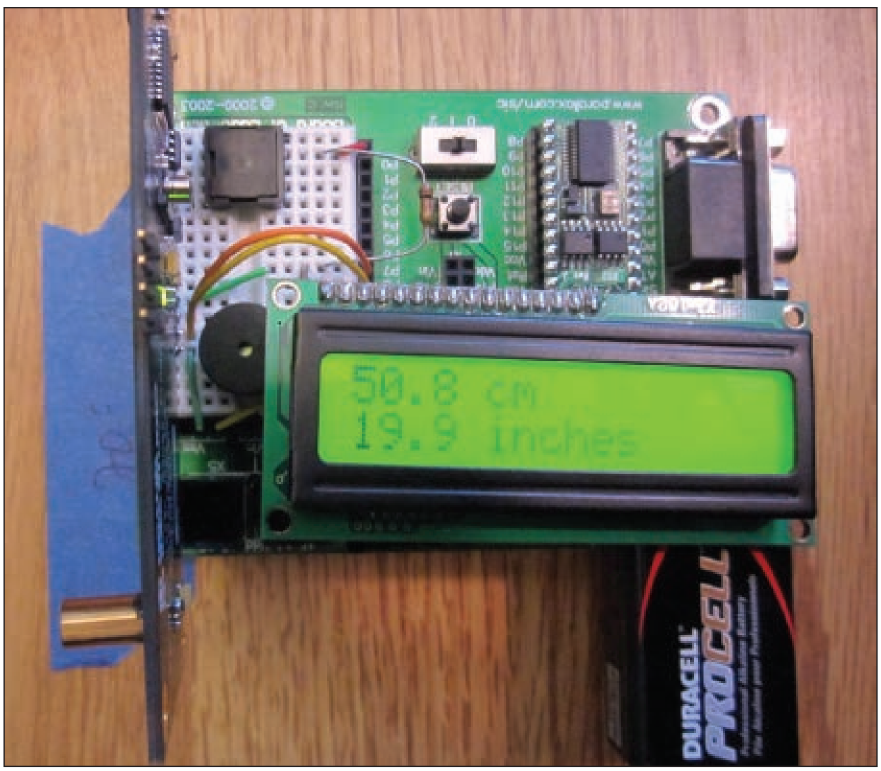


FIGURE 6. *Demo — BS2.*

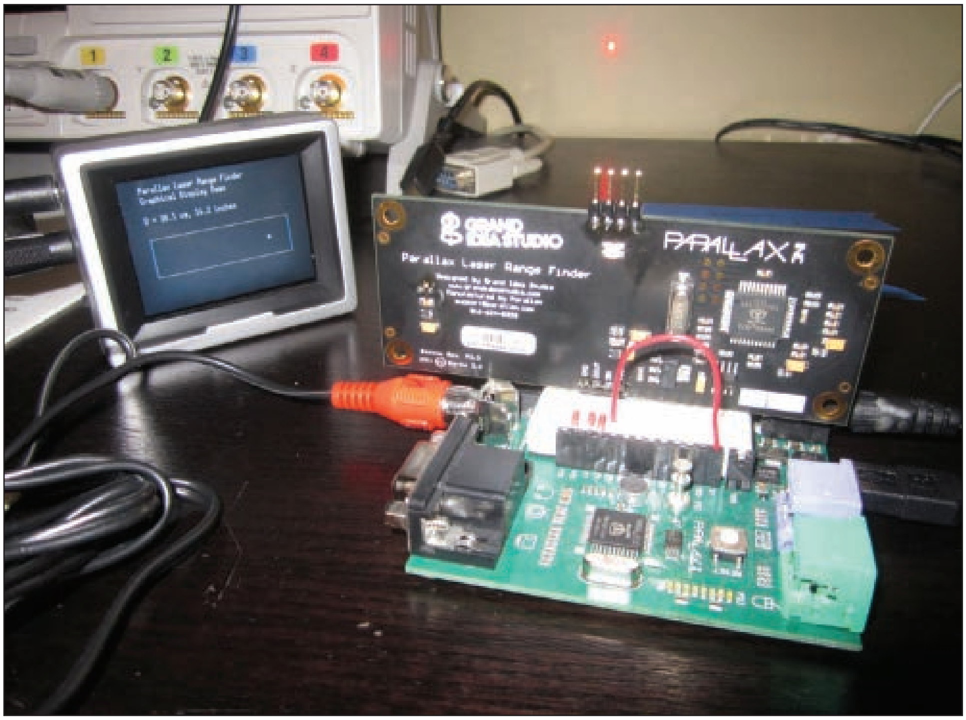


FIGURE 7. *Demo — Propeller Graphical.*

projects, I've put together a few simple examples:

- **Portable Laser Range Finder.** A portable unit using a BASIC Stamp II Board-of-Education, piezo buzzer, Parallax serial LCD module, and 9V battery (**Figure 6**). When the pushbutton switch is pressed, the LRF calculates the distance between itself and the target object. The result is displayed on the LCD in centimeters and inches. When the measurement is out-of-range (outside of the defined minimum or maximum distance bounds of the LRF), the piezo buzzer will sound a warning tone. Video: **www.youtube.com/watch?v=MqYRM6calOI**.

- **Graphical Interface.** Using the Propeller as a host (a separate Propeller, not the one onboard the LRF module), the distance between the LRF and the target object is displayed in text and graphical form via composite video output (**Figure 7**). Video: **www.youtube.com/watch?v=MqYRM6calOI**.

- **Proximity Sensing Robot.** Using Freescale's FSLBOT robot platform (**http://freescale.com/mechbot/**), the LRF module is mounted to the robot's head and interfaced directly to the MCF52259 ColdFire processor on the TWR-MECH board (**Figure 8**). The robot walks/waddles forward until it becomes too close to an object, then stops until the object goes away. Video: **www.youtube.com/watch?v=bvmT9h9ghk0**.

- **Host PC.** Since the LRF communicates via standard, printable ASCII, it can be controlled with any host computer's terminal program (e.g., HyperTerminal, PuTTY, or the Parallax Serial Terminal; **Figure 9**). The only additional hardware required is a USB-to-serial interface or level translator to properly interface the LRF's serial lines to the computer.

The LRF Image Viewer application (discussed earlier in this article) can also be used for easy interaction with the LRF module. Videos: **www.youtube.com/watch?v=MqYRM6calOI** and **www.youtube.com/watch?v=iHvMl2scUdA**.

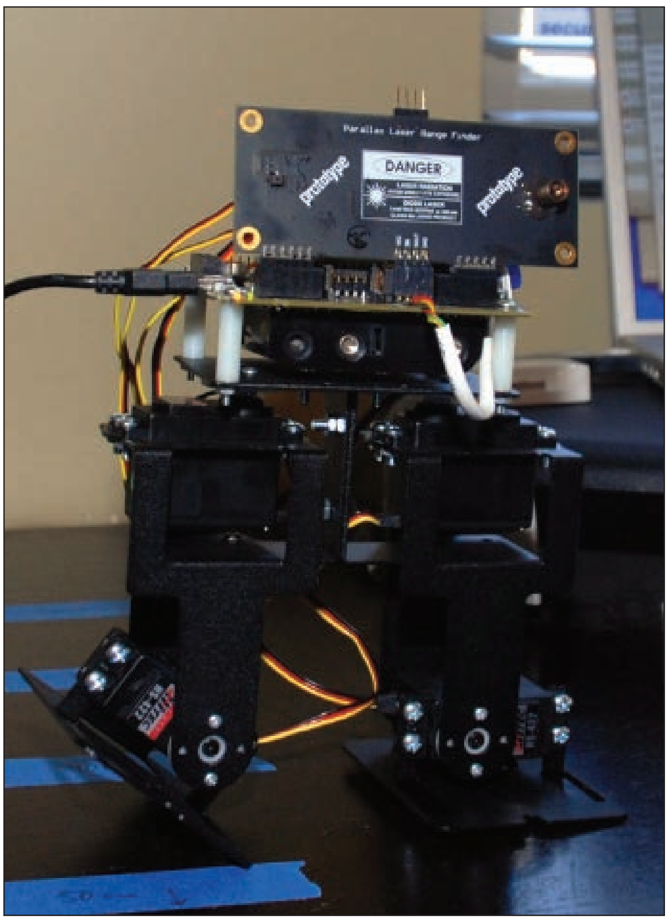
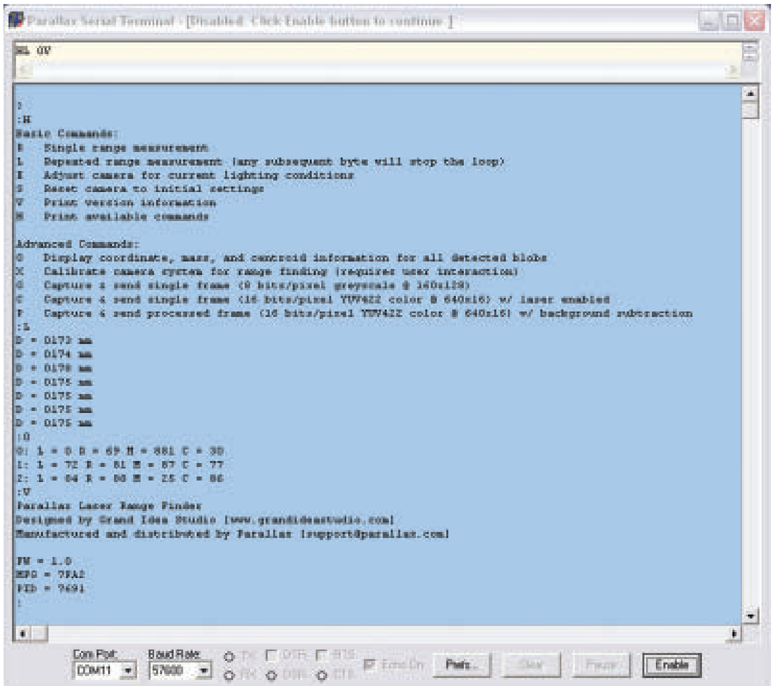


FIGURE 8. *Demo — FSLBOT.*

FIGURE 9. *Demo — Serial Terminal.*



Mission Accomplished!

Spanning nearly three years of research, experimentation, and design, I'm extremely satisfied with the outcome of the Laser Range Finder project. I've met my goals of creating a low cost, easy-to-use device and look forward to seeing what sorts of applications the LRF is used for. Since the entire project is released as open source, I also hope that my work is used as a starting point to create other interesting Propeller-controlled camera/vision systems. **SV**

Joe Grand is an electrical engineer and the president of Grand Idea Studio (**www.grandidea studio.com**), where he specializes in the design and licensing of consumer products and modules for electronics hobbyists. He can be reached at joe@grandideastudio.com.

Using Advanced Sensors with VEX — the VR Control Glove

By Daniel Ramirez

Virtual Reality (VR) is a growing field that is normally too expensive for the hobbyist of average means to afford to pursue, and is left for university, commercial, or military research labs (and, of course, video games). Mattel gave us the original Power Glove that worked as a video game input device, but stopped selling it in the late '80s. These power gloves sometimes can be found for sale on eBay.

Remember Mickey Mouse with the universe at his command in Disney's movie *Fantasia*? Well, imagine a glove that you wear that provides the ability to move objects towards you or away from you with the twist of a wrist. How about a glove you can use to steer an RC vehicle or robot in any direction just by pointing your hand in the desired direction. Now imagine what you could do with two such gloves. How about bringing some amazing animatronics to your own home for holidays or other special occasions?

Not too long ago — before the Wii controllers became widely available as motion input devices for new video games — it was joysticks, trackballs, or a mouse that were primarily used by video game players. This was also when the Mattel power glove came into play, using flexible resistors to digitize the operator's hand and finger movements.

Back in the August issue, I described the VEX sensor subsystem, including info on each of the available VEX sensors, how they correspond to our human senses (vision, hearing, feeling, touching, smell), and how they are used to provide feedback for controlling motor speed and detecting mechanical limits. I also showed you how to use sensors to control a simple but very bright strobe light. This time, I'll continue with this theme by describing other VEX resistive-based sensors used to simulate the sense of feel and touch, and will show you how to use the VEX construction set and these resistive-based sensors to build better robots. We'll start with the potentiometer.

Potentiometers come in all shapes and sizes as shown in **Figure 1**, and are used in commercial and consumer electronics from radios, TVs (the volume control), and joystick-based games. They're also used in rheostats in electric stoves and ovens to control the cooking temperature. In electronic circuits, potentiometers are used to implement variable voltage divider circuits. The rectangular shaped potentiometers are also known as "trim pots" and are usually used to finely adjust gains and offsets to electronic circuits.

The schematic symbol for a potentiometer is shown in **Figure 2**. Notice that a potentiometer has three terminals. The center terminal is connected to the wiper while the remaining two terminals are connected to the minimum and maximum voltages to be divided — zero volts and five

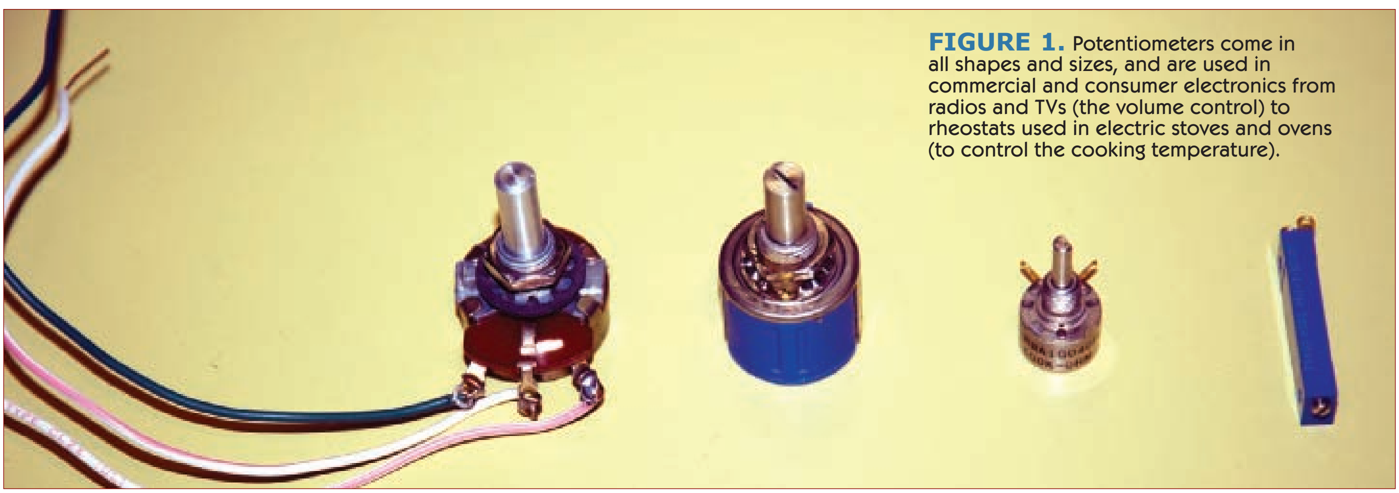


FIGURE 1. Potentiometers come in all shapes and sizes, and are used in commercial and consumer electronics from radios and TVs (the volume control) to rheostats used in electric stoves and ovens (to control the cooking temperature).

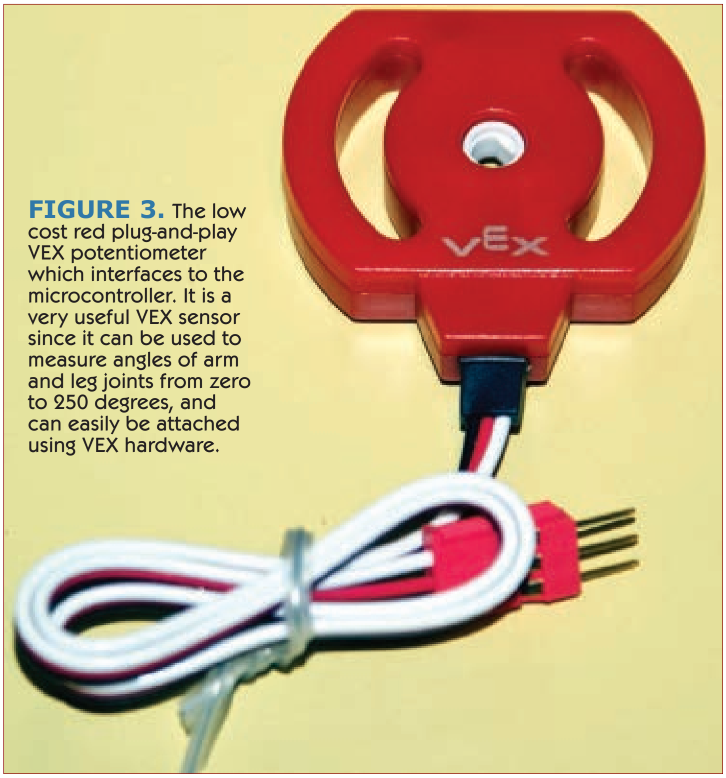


FIGURE 3. The low cost red plug-and-play VEX potentiometer which interfaces to the microcontroller. It is a very useful VEX sensor since it can be used to measure angles of arm and leg joints from zero to 250 degrees, and can easily be attached using VEX hardware.

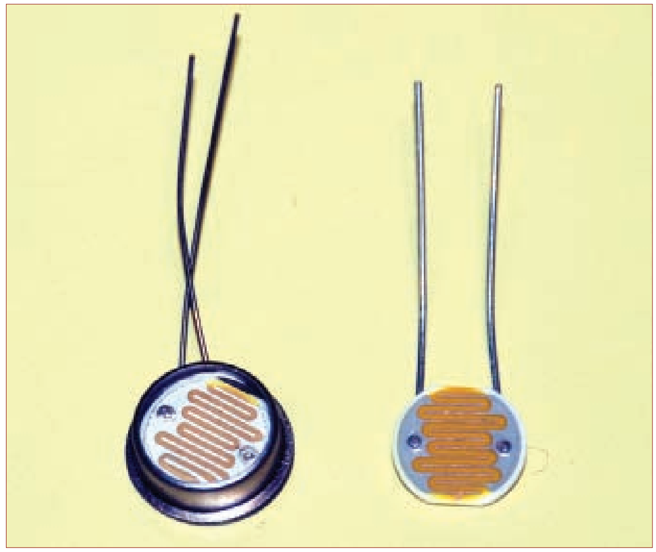


FIGURE 4. There are other kinds of resistive sensors that work similarly to potentiometers, including CDS cells (light sensors where the resistance varies proportionally to the brightness).

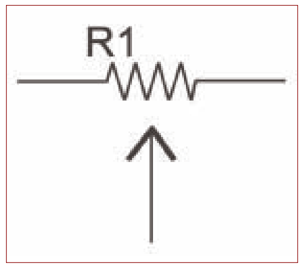


FIGURE 2. The schematic symbol for a potentiometer. Notice that it has three terminals.

volts, respectively, for most VEX applications.

The low cost red plug-and-play VEX potentiometer which interfaces to the microcontroller (sold by IFI) is shown in **Figure 3**. It is a very useful sensor since it can be used to measure angles of arm and leg joints from zero to 250 degrees, and can easily be attached using VEX hardware. Other applications for it include accurate speed control where the motor speed is proportional to the resistance or voltage at the center terminal (wiper) on the potentiometer. The August article mentioned that the potentiometer could be substituted for the quadrature optical encoder which was used as a strobe light frequency control. This time, we'll explore more of this sensor's capabilities and how it is similar to other resistive-based sensors.

There are other kinds of resistive sensors that work similarly to potentiometers, including CDS cells (light sensors where the resistance varies proportionally to brightness) shown in **Figure 4**, thermistors (resistors that vary with temperature), and a flexible resistor shown in **Figure 5** that is otherwise known as a flex sensor (a resistor whose resistance varies with the degree of flex or bend). Note that it only has two terminals and requires a 10K pull-up resistor to +5V in order to form a voltage divider. Since these sensors are all analog in nature, they can be used in a similar manner. These kinds of sensors are read using the microcontroller's analog inputs which are connected internally to the 10-bit ADC (Analog-to-Digital Converter). We will explore temperature and light sensors in a future installment. Now, journey with me to a virtual reality by carrying out the following VEX control glove experiment.

The VEX Control Glove Experiment

In this experiment, we will have some fun making our glove using flexible resistors which work very similarly to potentiometers. Take a look at the bill of materials (**Table 1**) needed to carry out this experiment. Note that you don't need to purchase all five flexible resistors; instead, you can keep the cost down and still get good results with one of these resistors (which cost around $13 each from SparkFun). This glove can easily be assembled in a couple of hours. I designed this glove for carrying out my own VEX-based Telepresence and telerobotics experiments. It can be used to liven up any magic show, musical production, or rock concert when the "star" uses it to cue the beginning of a dazzling light, laser, or pyrotechnics show to wow the

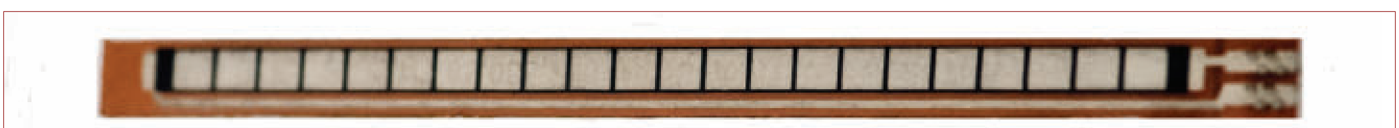


FIGURE 5. A flexible resistor otherwise known as a flex sensor (a resistor whose resistance varies with the degree of flex or bend). Note that it only has two terminals and requires a 10K pull-up resistor to +5V in order to form a voltage divider.

TABLE 1. **Bill of Materials needed for the VEX control glove.**

ITEM	QTY	DESCRIPTION	SOURCE
1	1	VEX microcontroller	Innovation First, Inc. (IFI) **www.vexforum.com**
2	1	VEX 7.2 volt battery	Innovation First, Inc. **www.vexforum.com**
3	1	Wire-wrap cable	RadioShack **www.radioshack.com**
4	1-2	*VEX potentiometers	Innovation First, Inc. **www.vexforum.com**
5	1-5	Flexible resistors	SparkFun **www.sparkfun.com**
6	1-5	10K resistors	RadioShack **www.radioshack.com**
7	1-5	*Pushbutton switches	SparkFun **www.sparkfun.com**
8	1-5	*LEDs	RadioShack **www.radioshack.com**
*Items are optional.			

audience. An excellent source of information for VR and the original Mattel Power Glove is Linda Jacobson's book, *Garage Virtual Reality* (Sams Publishing, January 1994).

I wanted to be able to experiment with a low cost, easy to make glove similar in function to the Mattel version which was originally sold as a motion input device for use with Atari and Nintendo arcade games. It was later customized by hobbyists to be used for their own VR applications.

VEX Control Glove Assembly Details

Our VEX control glove is shown in **Figure 6**. As mentioned, this version is low cost and very simple to assemble. We'll use a single flexible resistor hot-glued to the index finger and connected to the microcontroller using analog I/O pin 2. The VEX motor is connected to the motor 2 input. The glove itself is just a welder's glove with flexible resistors glued to each of the glove's fingers. We only use hot glue to mount the flexible resistors since it is also flexible when cool. Rubber bands can be used to hold the resistors in place while the glue cools. The wires are connected to the microcontroller as shown in the schematic in **Figure 7**. The additional resistors are assembled in the same manner and connected to their corresponding ADC pin on the microcontroller. Pushbutton switches are tiny PC or surface-mount switches that are glued to each of the glove's knuckles

Caution*: A glue gun temperature can reach 400 degrees Fahrenheit. Please be sure to wear goggles and a pair of working gloves while gluing the parts.*

Start assembling the glove by laying it flat, face down on a clean piece of cardboard or wood. Preheat your glue gun and apply a bead of hot glue along the middle of the index finger. Carefully lay one of the flexible resistors down on it while making sure that the metal pins face back towards the palm of the glove. Do not cover the metal pins with glue at this time because you will need to wrap two thin wires to each of the metal pins. Repeat this procedure for the remaining fingers. Also use the glue gun to mount the rest of the components.

The glove is wired and connected to the microcontroller using the **schematic**. Placement of the flexible resistors is not critical; they can just be hot-glued onto the center line of each of the fingers. Just make sure that the off-white rectangles on each strip are facing out. Each flexible resistor varies in resistance between 30K and 50K ohms when it is flexed to its extreme positions. Verify this fact using a digital multimeter before and after gluing the resistors in place in order to obtain the maximum sensitivity and response. Additional resistors are assembled in the same manner and connected to their corresponding ADC pin on the microcontroller as shown in the **schematic**.

LED Indicators

Optional LED indicators (also shown in the **schematic**) can be glued onto each finger or to the palm of the glove to provide a visual feedback to the wearer that a command has been issued. They also provide a simple menu and are used during the glove calibration process, indicating when each finger has been calibrated. The glove switches can also have LED indicators that show the toggle state of each switch. There is no need for current-limiting resistors since the VEX microcontroller digital outputs already have them internally. Just insert an LED into any unused digital output, connecting the anode to the selected digital output and the cathode to ground. No connections should be made to the

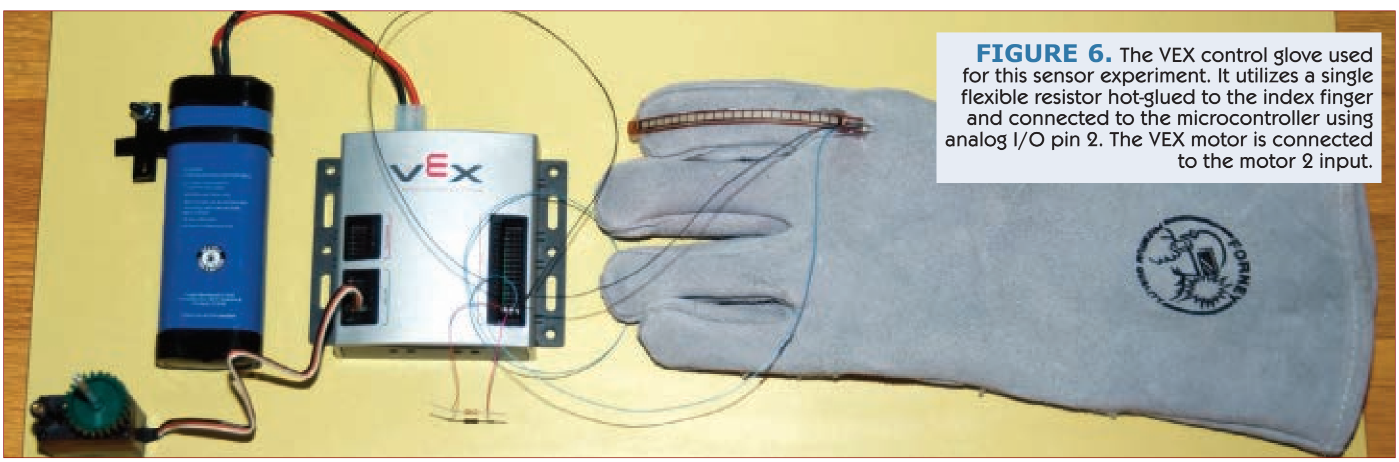


FIGURE 6. The VEX control glove used for this sensor experiment. It utilizes a single flexible resistor hot-glued to the index finger and connected to the microcontroller using analog I/O pin 2. The VEX motor is connected to the motor 2 input.

middle socket since it is the +5 volts. Modify the firmware to turn the LED on or off, depending on the pushbutton pressed. For this experiment, just insert a green LED in IO11, as a calibration mode indicator and a red LED into IO12 as a run mode indicator.

Wire Harness

Signals from the flexible resistors are routed to the microcontroller using a tether made from thin and light flexible wire-wrap or stranded wires. Thin stranded wire is preferable for the fingers since they include a lot of flexing motions that could break the wire-wrap or solid wire with continuous use. The power, ground, and signal wires leading from each of the flexible resistors, sensors, accelerometer boards, and pressure sensors are carefully routed to the control box via the cable harness. Connect the ground terminals of each of the switches and flexible resistors together, and route the single ground wire into the wire harness towards the back of the glove. Use plastic ties to group and hold the harness wires in place. Be sure to carefully label each of the wires leading from the sensors and switches so that they can be connected to the corresponding I/O pins without mixing them up.

Insert the wire harness into a shrink wrap tube but do not shrink it at this time. The wires leading from each of the fingers via the glove's wire harness and pushbuttons are connected to the microcontroller.

To weatherproof the glove, you can encapsulate the wire harness with shrink-wrap tubing to protect the delicate wires from damage (once you are sure that everything works). Use a hot air gun to shrink the tubing. The wires leading from the flexible resistors should be carefully routed towards the back of the glove and inserted into the shrink-wrap tubing. The harness is attached to the glove (also leading to the back) using beads of hot glue. Plastic ties can be utilized to hold the wires in place if necessary.

For outdoor use, enclose all the glove's electronics and battery power supply in a plastic or metal project box. Make a hole in the box large enough to pass the wire harness and shrink-wrap tube through and anchor it inside the project box with hot glue.

Glove Position

I won't be discussing the glove's position data this

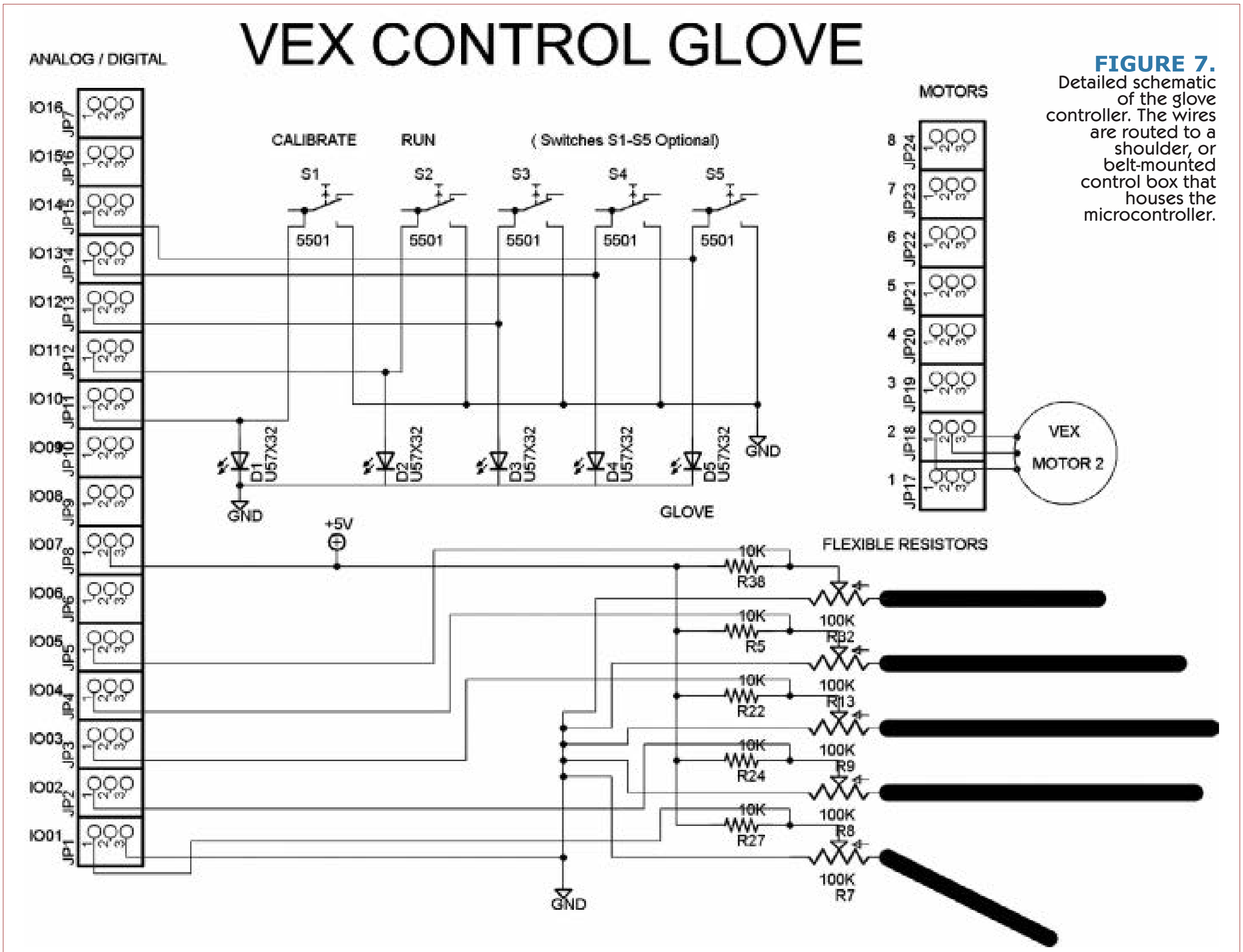


FIGURE 7. Detailed schematic of the glove controller. The wires are routed to a shoulder, or belt-mounted control box that houses the microcontroller.

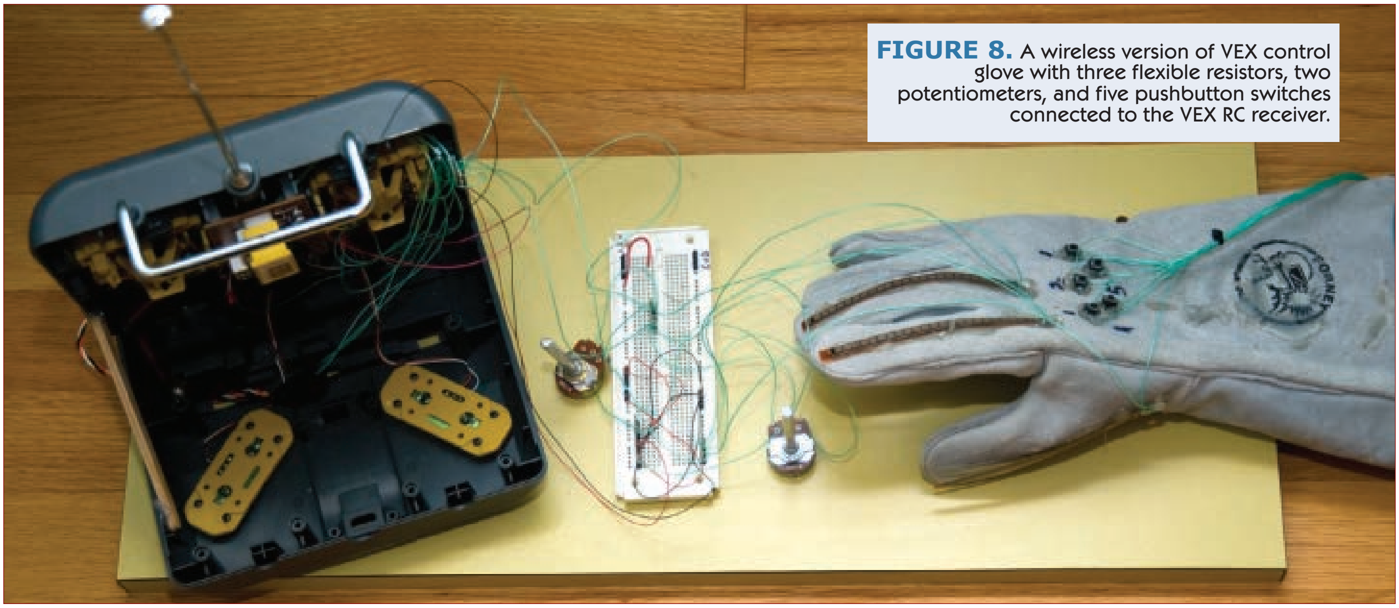


FIGURE 8. A wireless version of VEX control glove with three flexible resistors, two potentiometers, and five pushbutton switches connected to the VEX RC receiver.

time. I do plan to obtain its position indoors by using a combination of ultrasonic PING))) sensors or Polaroid 6500 sonar rangers with Time of Flight (TOF) and trilateration or triangulation techniques to calculate the glove's position in 3D. Another alternative would be to implement a low cost gyro and accelerometer combination board (IMU) used for RC helicopters to keep track of the glove's position and orientation.

Finger Positions

The control glove's fingers allow you to move distant objects by just bending your index finger with natural motions. The finger positions are read using the microcontroller's 10-bit ADC to digitize the voltages that are generated by bending the flexible resistors; this changes the resistance depending upon how much each resistor is bent. In order to obtain the best readings possible, each finger is individually calibrated by having the operator move it from its minimum position to its maximum position, then the voltages are recorded so that they can be used to calibrate the finger position data. The finger positions are mapped by the remote application using the calibration data so that each finger position can move the individual servo or motor (or multiple servos or motors).

FIGURE 9. A wireless glove is the perfect companion to controlling VEX robots such as the Telepod tripod (photographer's assistant).

Pushbuttons

Five small pushbuttons (shown in **Figure 8**) provide functionality similar to a PC joystick trigger and fire buttons, but also allow the wearer to select which motors to move or which lights to turn on/off. The pushbuttons can be read and de-bounced using the Easy C Pro digital input function firmware for the IO11-IO16 digital input pins. Pushbutton switches are of the miniature PCB variety. Each is hot glued to each forefinger or knuckle so that it is easily accessible from the user's other hand. Each pushbutton switch on the glove is wired to the digital inputs IO11 through IO16 of the microcontroller using wire-wrap wire.

The pushbutton switches can be assigned to perform a task on the microcontroller as shown in **Table 2**.

How It Works

The system consists of the microcontroller and glove to digitize the finger positions (flexible resistors) and the glove's orientation. It works by having the Easy C Pro firmware simply digitize the glove's finger positions by reading the voltages across the flexible resistors whose resistance varies with the angle that each finger is bent or flexed. The wearer's own finger movements are digitized

TABLE 2. Pushbutton switches.

BUTTON	ANALOG/DIGITAL PIN	FUNCTION
1	IO11	CALIBRATE
2	IO12	RUN
3	IO13	RECORD
4	IO14	PLAY
5	IO15	FIRE

```
// Stop motors
SetPWM ( 1 , 127 ) ;
SetPWM ( 2 , 127 ) ;
SetDigitalOutput ( 11 , 1 ) ;
// Turn on Calibration LED
SetDigitalOutput ( 12 , 0 ) ;
// Turn off RUN LED
PrintToScreen ( "Calibrate Finger 2 (index
finger)...\n" ) ;
PrintToScreen ( "Calibrate Finger 2 (index
finger)...\n" ) ;
Wait ( 5000 ) ;
for ( i=0 ; i<NUMBER_OF_CYCLES ; i++ )
// Calibrate for NUMBER_OF_CYCLES
{
        Finger_2 = GetAnalogInput ( 2 ) ;
        if ( Finger_2 < Min_Finger_2 )
        {
                Min_Finger_2 = Finger_2 ;
        }
        if ( Finger_2 > Max_Finger_2 )
        {
                Max_Finger_2 = Finger_2 ;
        }
    PrintToScreen ( "i = %d" , (int)i ) ;
    PrintToScreen ( " Finger 2 = %d" ,
    (int)Finger_2 ) ;
    // Display value of finger 2
    PrintToScreen ( " Min Finger 2 = %d" ,
    (int)Min_Finger_2 ) ;
    // Display value of finger 2
    PrintToScreen ( " Max Finger 2 = %d\n" ,
    (int)Max_Finger_2 ) ;
}
PrintToScreen ( "Calibration for Finger 2 (index
finger) complete!\n" ) ;
Wait ( 5000 ) ;
SetDigitalOutput ( 11 , 0 ) ;
// Turn off Calibration LED
SetDigitalOutput ( 12 , 1 ) ; // Turn on RUN LED
```

LISTING 1. This Easy C Pro example shows the calibration mode.

```
set_view (Min_Finger_2 , Min_Finger_2 ,
Max_Finger_2, Max_Finger_2 );
// Set viewport to calibrated minimum and
// maximum finger 2 positions
set_window (127.0, 127.0, 255.0, 255.0);
// Set the window to scale motor commands
// between 127 and 255
while ( 1 ) // VEX Control Glove Calibration
loop
{
        Finger_2 = GetAnalogInput ( 2 ) ;
        map_viewport_to_window (Finger_2 ,
        Finger_2 , &Motor_2, &Motor_2 );
        // Map finger 2 positions to motor 2
        commands
        Motor_Command_2 = (int) Motor_2;
        PrintToScreen ( "Finger 2 = %d" ,
        (int)Finger_2 ) ; // Display value of
        finger 2
        PrintToScreen ( " Motor 2 Command = %d\n"
        , (int)Motor_Command_2 ) ;
        // Display value of finger 2
        SetPWM ( 2 , Motor_Command_2 ) ;
        // Send motor command to motor 2
}
```

LISTING 2. This Easy C Pro example shows the run mode.

using the 10-bit ADC and are processed and filtered by the firmware. The10-bit reading from the ADC for any of these sensors is a value between zero and 1023 that can be scaled to PWM motor commands needed for animation. Calibration data consisting of the minimum and maximum digital value representing the voltage read is necessary to scale the VEX motor commands. Once the application is in the RUN mode, watch the motor speed up or slow down or stop, depending on the current angle of your index finger.

Wireless VEX Control Glove

A more advanced wireless version of the glove is made by connecting it to a VEX RC. In this version, we use three flexible resistors, two potentiometers, and five pushbutton switches (shown in **Figure 8**). This version is able to control VEX robots remotely. The flexible resistors and potentiometers provide precise motor speed control from a safe distance. I purchased a low cost surplus VEX RC unit for this experiment on eBay.

A wireless glove is the perfect companion to controlling robots such as the Telepod tripod (photographer's assistant) shown in **Figure 9** since it can be used as the motion input device for it (using the RC). We can also use the glove to interface to a laptop or desktop PC, or even a graphics LCD so that a real time display of the glove's position and orientation is shown to the user. Used in this manner, you can generate the simulated glove's positions to be used with PC video games. This advanced glove uses various sensors including flexible resistors, XYZ accelerometers, pressure sensors, temperature sensors, and pushbuttons to enable the wearer to control almost any type of animatronic prop or robot (including VEX-based models). For instance, each finger could be assigned a specific motor or servo that is attached to the prop. By tilting your hand, you can turn a remote controlled car or robot left or right. Bending the index finger speeds it up or straightening it can slow down or stop a car, for example. I'll describe how this was done in a future article if there is enough interest.

VEX Control Glove Applications

This glove has many applications other than just controlling the speed or position of a motor or servo. It can be used as a motion input device or "teaching pendant" to train your robot in a natural manner so that it can repeat moves autonomously when required to do so. When connected to an RC, not only does this glove provide the wearer the power to animate objects at a safe distance, it allows the user to experiment with VR in applications such as:

- Robots
- Props/animations
- Special effects at magic shows, laser shows, musicals, and rock concerts (conductor's baton)
- Photography

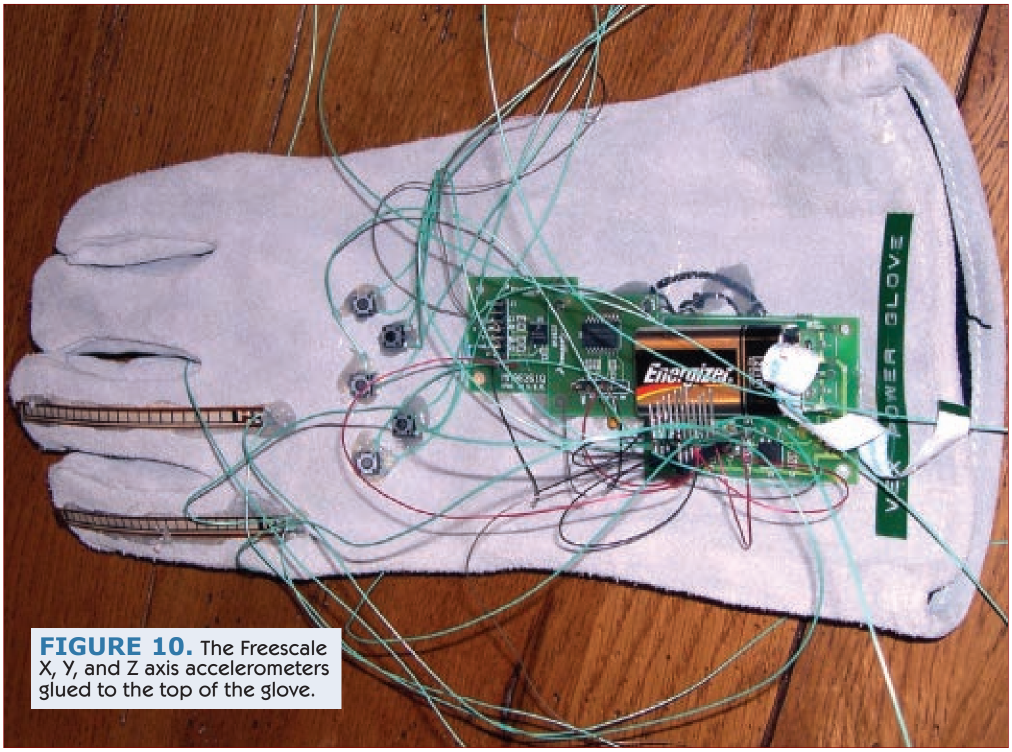


FIGURE 10. The Freescale X, Y, and Z axis accelerometers glued to the top of the glove.

Safety

A safety issue can arise because of the glove wearer's ability to twitch or inadvertently move his/her hand or fingers, causing a rapid position shift of the glove which results in a corresponding rapid movement of the robot or prop. For this reason, keep yourself, and bystanders, and any children away from the prop or robot's workspace so that the remote controlled prop or robot does not accidentally injure anyone. Even a toy RC car can cause injury if traveling fast enough.

Firmware

The EasyC Pro application needed to read a single flexible resistor and control the motor speed (or position proportional to the degree that the finger with the flexible resistor is bent) is named VexControlGlove. It and the complete source code is available from the article link shown.

Calibration Mode (Green LED)

In order to use the glove effectively, it must be calibrated periodically to provide accurate finger positions and glove orientations. For this experiment, the calibration mode shown in **Listing 1** is automatically run each time the microcontroller is reset, or it is run from the Easy C Pro IDE. Just after resetting the microcontroller, the green LED indicating that the glove needs to be calibrated will turn on. The calibration process involves just flexing each finger to the extreme open and closed positions while the green LED is on. For this experiment, we only use the index finger, so it needs to be calibrated. If you add more flexible resistors, then you need to modify the firmware to calibrate them in the same manner.

I only implemented up to three fingers to keep the cost down, but depending on your resources, you can add two more flexible resistors on the right glove and also add up to five flexible resistors to a left hand glove, if desired. The calibration procedure is the same for the remaining fingers of a right or left hand glove.

Run Mode (Red LED)

Once the calibration mode is complete, the green LED turns off and the red LED turns on, indicating the glove firmware is in the run mode (shown in **Listing 2**).

Calibration data is used to compute the gains and offsets necessary to scale or map the flexible resistor readings to actual motor commands using the **map_viewport_to_window** function provided with the source code.

VEX Control Glove Improvements

The glove described here is just the tip of the VR iceberg. There are many improvements that can be made to make it even more responsive and functional. The original Mattel Power Glove used an ultrasonic 3D ranging device to determine the glove's position and orientation, but this feature can be expensive and very advanced for most VEX users (including myself).

New MEMS sensors have provided lower cost alternatives using the VEX XYZ accelerometers and new gyros (IMU) to provide the glove's XYZ axis tilts and pitch, yaw, and roll angles, along with the ability to integrate the accelerations in each axis to obtain its position. This is like RC helicopters.

SparkFun (**www.sparkfun.com**) sells IMU combination boards that have this functionality for example, the SEN-09268 IMU analog combo board with five degrees of freedom. The Freescale XYZ accelerometer shown in **Figure 10** is glued to the back of the glove, directly opposite of the palm. Care must be taken to align the sensor so that it is correctly centered and aligned with the X, Y, axis.

Tactile feedback from the glove can be obtained using new Freescale pressure sensors (also sold at SparkFun) that are glued to the tips of the fingers. Each of the pressure sensor outputs are connected to the microcontroller's analog inputs so it can read the pressure between the finger and the object it is holding. SV


mymindsi.com
BUILD ANYTHING
imagine it
explore it
build it
play-it
program it
Ignite your creativity and imagination with the all new MINDS-i robotics system.
Finally, a rugged quick-lock construction system that empowers you to envision, build, and re-create anything that you can imagine with your own "mind's eye".
DO ANYTHING
MINDS-i
PCS ROBOTICS
Utilize PCS Robotics controllers, sensors, and software to bring MINDS-i to life. Use PCS curriculum in the classroom to elevate your student's interest in science, technology, engineering and math!
Patents US 7,517,270; US 7,410,225 B1; US 7,736,211; US 7,841,923; International Patents Pending.
pcsedu.com/servo


The Robot MarketPlace
Have a developing robot enthusiast on your gift list?
CHALLENGE YOUR MIND
with educational robotics kits for all skill levels.
• Surveillance Robots
• Combat Robots
• Full Line of R/C Cars, Trucks Boats, Airplanes, & Helicopters
• Hobby R/C Accessories
Visit us online to view over 11,000 robot parts and hobby accessories!
(877) ROBOT 99
(877-762-6899)
Your Headquarters for R/C Hobby & Robot-Related Gifts!
www.RobotMarketPlace.com

Beginner Bot Meets Arduino

Part 4

by Gordon McComb

For the last several issues, you've read about the Beginner Bot: an affordable and expandable platform that demonstrates basic robotic concepts. You've learned how to construct the Beginner Bot platform using wood or plastic, and how to steer it using mechanical switches from a tethered control panel.

*Important note! This article relies on construction details described in the earlier parts of this series. While you don't absolutely need to build each stage of the Beginner Bot in the sequence outlined in this series, if you're just starting out you'll want to refer to the earlier articles. Links are available at the article download page at **www.servomagazine.com**. There you'll learn from the ground up how to construct, wire, and use the Beginner Bot.*

From there, you learned how to replace the switches with fully electronic influence, adding twin "eyeballs" and a simple one-chip brain so that your bot can follow the beam of your flashlight.

You then discovered how to interface the Beginner Bot to a PICAXE microcontroller, replacing hard-wired control circuitry with software programming. The Beginner Bot isn't tied to any one microcontroller, so this time around I'll show you how to connect and program an Arduino Uno development board to run your robot in circles — literally! The finished Beginner Bot as described in this article is shown in **Figure 1**.

Crash Course in Arduino

New to the Arduino? Then, be sure to read this section which summarizes what it is and how it's used. Feel free to skip this part if you're already familiar with it.

The Arduino is a development platform, meaning that it combines a microcontroller chip with a circuit board and other hardware for ready experimentation. **Figure 2** shows the Uno, one of several Arduino boards. It's the most popular version, and is the one used in this month's project. It features a low cost Atmel ATmega328P microcontroller IC mounted on a handy "stackable" board. The board itself measures 2-1/8" by 2-3/4", and is the same form factor as the PICAXE 401 development board we used last month.

Main points of interest of the Arduino Uno are:

- Reset pushbutton: Press to reset the currently running program.
- Integrated USB-to-serial communications for both uploading programs from your PC and for serial communications back to the PC for debugging and monitoring. The USB link includes a

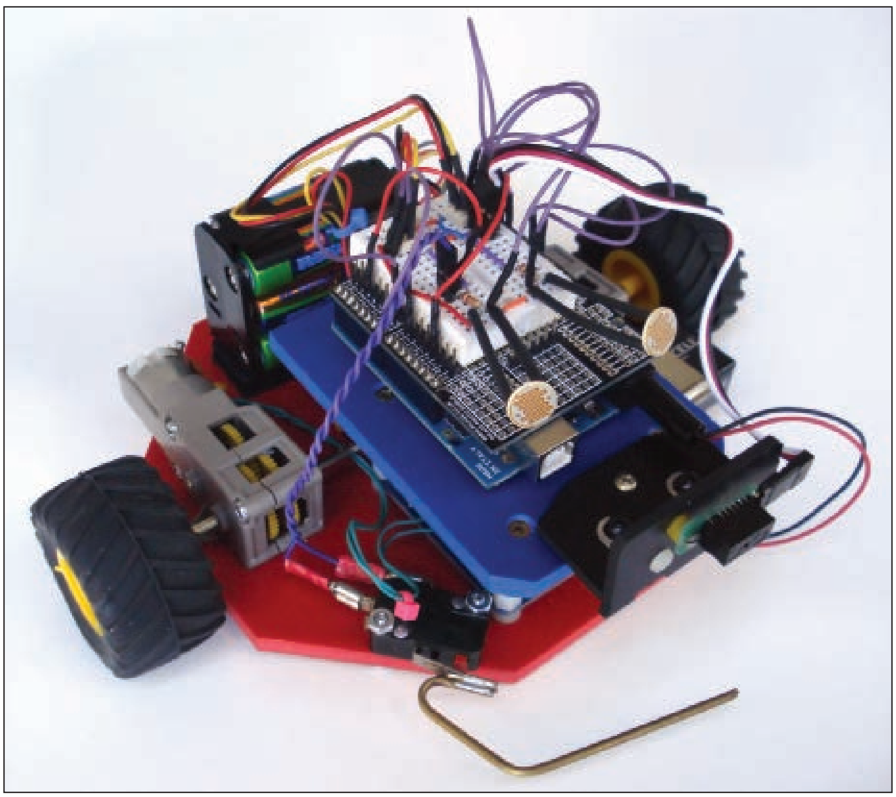


FIGURE 1. The completed Phase 4 version of the Beginner Bot, with Arduino microcontroller board, prototype shield with mini solderless breadboard, and various sensors and wiring.

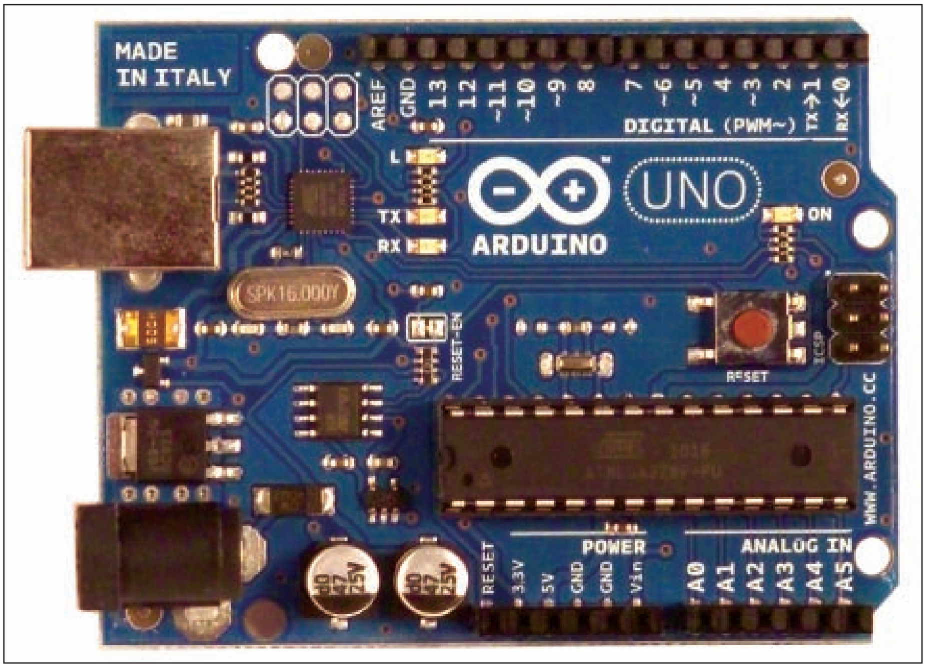


FIGURE 2. Arduino Uno development platform board.

500 mA resettable fuse to guard against possible damage caused by a wayward Arduino to the USB ports on your PC. When plugged into a USB port, the Arduino takes its power from it.

- DC power jack (2.1 mm, center positive) for use with an external power source. Recommended voltage range is 7-12 volts.
- Low dropout regulators for 5V and 3.3V. The five volt regulator provides up to 800 mA of current; the 3.3 volt regulator provides 50 mA. Connection pins are provided for both the 5V and 3.3V regulated outputs. You can use these pins to power low current components such as infrared sensors.
- Indicator LEDs for power, serial transmit, and receive pins, and digital pin 13 (labeled L).
- Six analog input/output (I/O) pins and 14 digital I/O pins. The analog pins connect to an internal 10-bit analog-to-digital converter, letting you read voltages from sensors and other devices. All I/O pins can be used as digital outputs.
- Power pins to provide external access to the unregulated and regulated power supplies. For the Beginner Bot, we only use regulated power.

As with nearly all microcontrollers, you program them by writing software on your PC, then upload the software to the Arduino development board. Arduino is no different, except that it refers to its programs as sketches. Sketches are written in a programming language very similar to C, but with some simplifications to aide newcomers. The Arduino comes with its own integrated development environment (IDE) software, available as a single download from the main **www.arduino.cc** website.

When you upload a compiled sketch to the Arduino, it is stored in 32K bytes of Flash memory inside the ATmega328 chip. Programs are stored permanently until you replace them with another sketch. The Arduino supports 1K bytes of electrically erasable non-volatile EEPROM (data survives after power-down) and 2K bytes of RAM. Data in RAM is volatile; it's lost when power is removed from the Arduino.

Mounting the Arduino

Part 2 of this series showed how to mount a small expansion deck to the Beginner Bot so you can easily add control electronics. The deck has enough space for the Arduino development board, with room to spare. Check out Part 3 of the Beginner Bot series for the full construction details, but here's a summary.

1. Remove the second deck (from Part 2) from the Beginner Bot and drill two holes to mount the Arduino. The Uno board has four mounting holes; pick the ones near the upper right and bottom left of the board. Center the board on the deck, leaving a little more space along the front. Mark the position for the two holes, and carefully drill using a 1/8" drill bit.
2. Use a pair of 4-40 jackscrews, nuts, and 4-40 x 1/4" machine screws to mount the Arduino. Jackscrews are like miniature standoffs, with male threads on one end and female on the other. If the jackscrews and other hardware are metal, add plastic washers to prevent a possible short circuit. The washers may not be needed, but they're good insurance.
3. Once the Arduino board has been mounted, you can attach the prototyping shield (see Part 3 for details) on top, with the mini (170 tie-point) solderless breadboard in the middle. You can securely attach the mini breadboard with self-adhesive foam tape or Velcro™. For my prototype, I soldered in some extra header pins front and back to keep the board in position.

Use the Right Motors!

The Beginner Bot uses a pair of Tamiya gearboxes that have been modified according to instructions provided in Part 2 of this series. Specifically, the motors used in the gearboxes have been replaced with versions that provide for operation at six to 12 volts, and with higher efficiency. These motors are available from Pololu (item #1117), among other sources. Cost is under $2 each.

Be sure to *not* use the stock motors that come with the Tamiya gearboxes. These are rated for only three volts and can consume copious amounts of current. This current exceeds the rating of the L298 H-bridge used to control the motors.

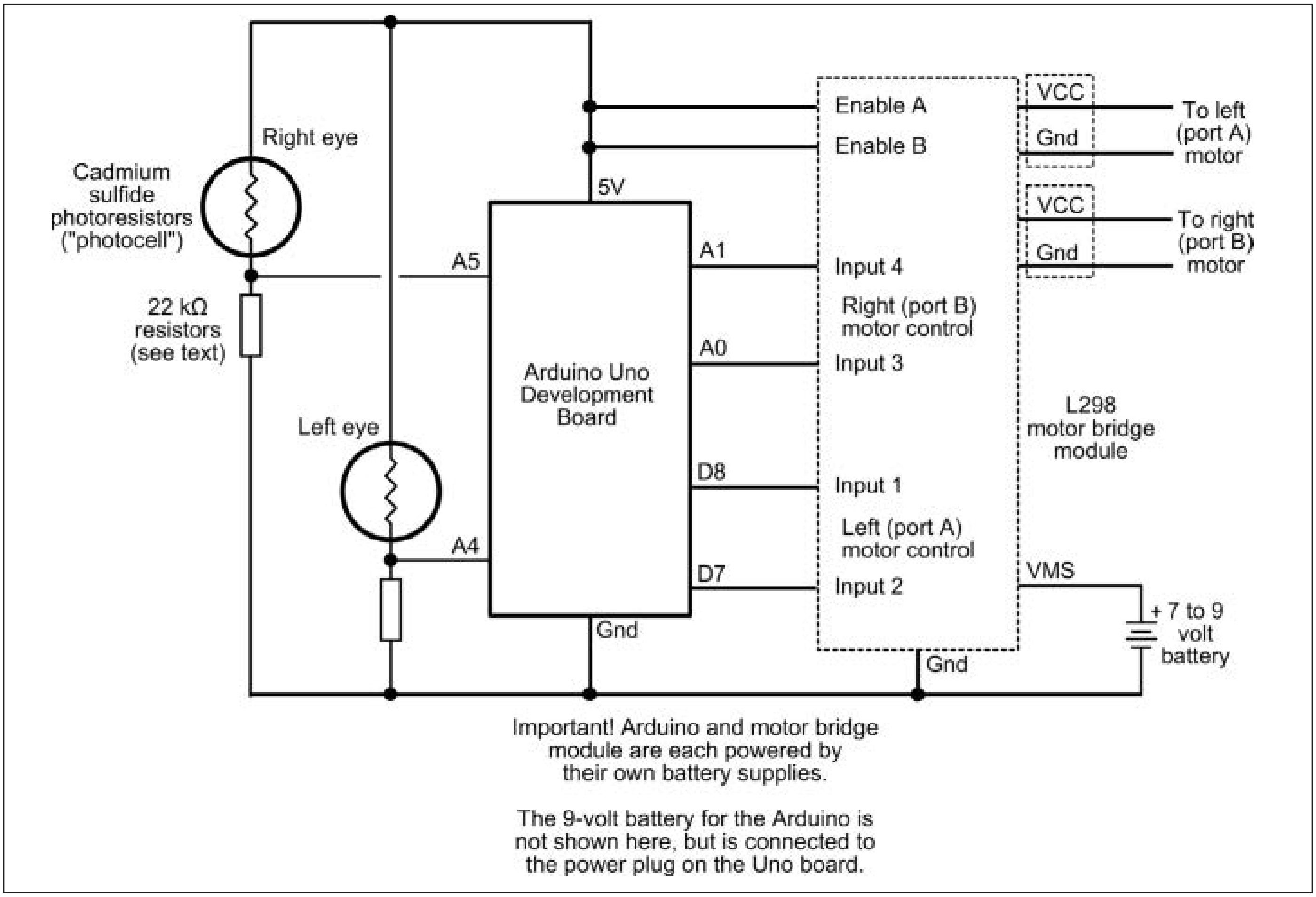


FIGURE 3. Schematic view of the CdS photocells, Arduino, and L298 motor driver board. Note that unlike the PICAXE version described last month, the Arduino does not derive its five volt power from the motor driver. Instead, the Arduino is operated from its own nine volt battery, connected to the barrel power plug.

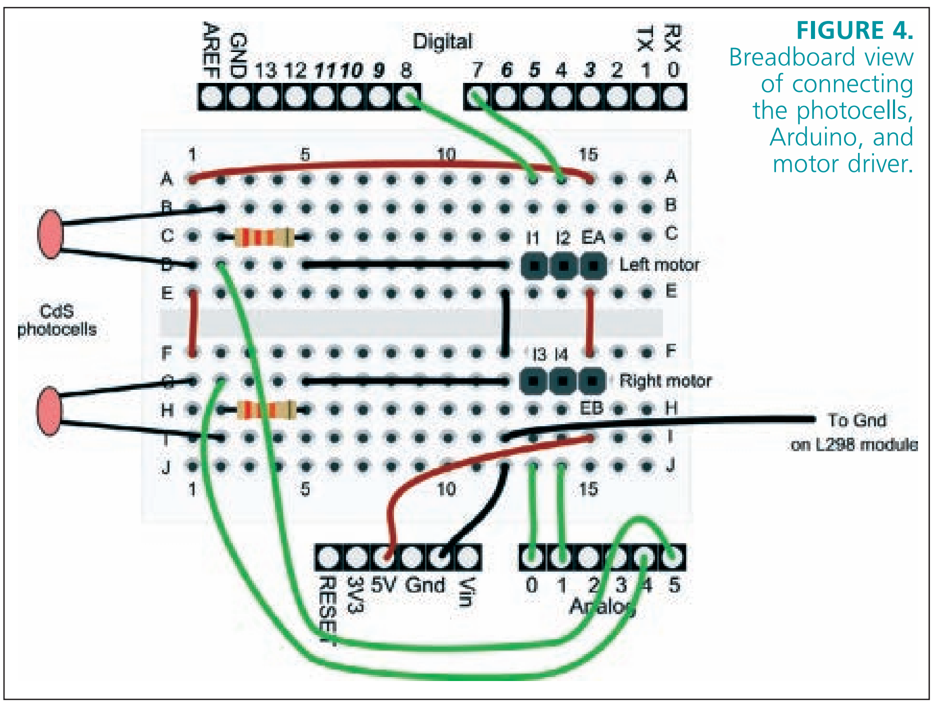


FIGURE 4. Breadboard view of connecting the photocells, Arduino, and motor driver.

Wiring for Motor Control

Phase 1 of the Beginner Bot project used switches for manual control of the bot's motors, and that was followed with converting to electronic control using an H-bridge module. With the Arduino, you can connect some of its input/output pins to the H-bridge module and operate the motors using software commands. You can easily modify the behavior of the Beginner Bot simply by altering a few lines of code.

Refer to Part 2 on how to mount the H-bridge to the Beginner Bot; connect the motors and wire a battery pack to it. I recommend using a six-cell AA battery holder and rechargeable nickel-metal hydride (NiMH) batteries. Keep the batteries freshly charged, as lower than normal supply voltages can

TABLE 1

Input A	Input B	What Happens
Low	Low	Motor stops
Low	High	Motor turns one direction
High	Low	Motor turns the other direction
High	High	Motor stops

cause mysterious and hard-to-track problems.

See **Figure 3** for how to connect the Arduino development board to the H-bridge module. Use the mini breadboard as an interface between the motor bridge, the two photocells, and the Arduino. **Figure 4** shows the circuit in breadboard view.

Important!! This wiring diagram is different from the one used last month with the PICAXE. Even though both the Arduino Uno and PICAXE 401 development boards share a similar pin-out arrangement, some of the connection points have been modified to simplify future expansion when using the Arduino.

Most importantly, in this version the Arduino does not derive its power from the 5V regulator on the H-bridge module. I found this caused some inconsistent behavior, so instead the Arduino is powered from its own nine volt battery. A common ground connection is used between the Arduino and the H-bridge module.

There's room on the base or back of the expansion deck for placing the nine volt battery. Use a piece of Velcro to secure it in position. Make or purchase a battery cable that has a standard nine volt battery clip on one end and a 2.1 mm (center positive) barrel plug on the other. The cable should be about 6" long. Avoid the kind with heavy duty wire or big fat insulation around the plug. You don't need it, and the extra bulk can get in the way.

The basic control circuit is pretty simple: two cadmium sulfide (CdS) photocells detect the amount of light falling on them. The photocell exhibits a change of resistance depending on the amount of light. The less light, the higher the resistance; the more light, the lower the resistance. For each CdS "eye," a 22 kΩ resistor turns the resistive output to a varying voltage. The resistance of the CdS cell — plus the fixed resistor — form a voltage divider circuit.

The two CdS photocells are mechanically attached to the front of the mini solderless breadboard (see **Figure 3** again). Gently bend the leads of the cells so that they point slightly upward and outward. Add heat shrink tubing (unshrunk) over the photocell leads to provide both insulation and added mechanical support.

The voltage produced by the CdS sensors stretches from between zero and five volts, and is connected to two of the Arduino's analog inputs — the pins marked

LISTING 1 – BegBotDrive.pde

```
const int waitDelay = 2000; // Delay between motion routines
const int in1 = 7;          // Input1 pins
const int in2 = 8;
const int in3 = A1;         // Input 2 pins
const int in4 = A0;

const int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);  // Set up pins as outputs
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // Indicate proper operation
  digitalWrite(ledPin, HIGH);
  delay(500);
  digitalWrite(ledPin, LOW);
  delay(500);
  digitalWrite(ledPin, HIGH);

}

void loop() {

  goFwd();                  // Go forward
  delay(waitDelay);         // Delay before next action

  goRev();                  // Reverse
  delay(waitDelay);

  goRight();                // Turn right
  delay(waitDelay);

  goLeft();                 // Turn left
  delay(waitDelay);

  allStop();                // Stop
  delay(waitDelay);

}

// Motion routines
void goFwd() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void goRev() {
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
}

void goRight() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
}

void goLeft() {
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void allStop() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
```

LISTING 2 – BigBotLightsteer.pde

```
Listing 2 - BigBotLightsteer.pde
// CdS cell reference values
//  (you need to experiment)
const int ambient = 600;
const int threshold = 800;

const int waitDelay = 2000; // Delay between motion routines
const int in1 = 7;          // Input1 pins
const int in2 = 8;
const int in3 = A1;         // Input2 pins
const int in4 = A0;

int lightLeft = 0;
int lightRight = 0;

void setup() {
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void loop() {
  // Read light sensors connected to
  //  analog pins A4 and A5
  lightLeft = analogRead(A4);
  lightRight = analogRead(A5);

  // Stop robot if below ambient
  if (lightRight < ambient || lightLeft < ambient) {
    allStop();
  } else {
     goFwd();
    // Steer to Left if right CdS below threshold
    if (lightRight < threshold) {
      goLeft();
      delay (250);
    }
    // Steer to right if left CdS below threshold
    goFwd();
    if (lightLeft < threshold) {
      goRight();
      delay (250);
    }
  }
}

[duplicate "motion routines" from Listing 1 here]
```

A4 and A5. The value of 22 kΩ for the resistors connected to each CdS cell is determined by trial and error. You may want to try different values to find the best sensitivity for the photocells you're using. You want the highest sensitivity while maintaining the widest possible swing between zero and five volts.

As noted in Part 2, the Seeedstudio H-bridge module — which uses an L298 motor control IC — requires at least two inputs per motor. The direction of the motor is determined by the instantaneous value of these two inputs, according to **Table 1**.

You control the operation and direction of either motor by setting the pins LOW (zero volts) or HIGH (five volts). You'll see exactly how this is done next.

Testing the Motors

On any new robot I build, I like to run through a series of motor movement tests to be sure everything is working. Refer to **Listing 1** for a demonstration program for checking the basic operation of the Arduino board, the H-bridge, and the motors. Type or download this program from the *SERVO* website into the Arduino IDE, then:

1. Place a small block under the Beginner Bot base to lift the wheels off your worktable.
2. Connect the battery to apply power to the H-bridge. You can leave the nine volt battery to the Arduino disconnected for the time being.
3. Plug in the programming cable between your PC and the Arduino, and be sure its communication port is selected in the Arduino IDE (choose Tools>Serial Port).
4. Compile the sketch by clicking the Verify button.
5. Load the sketch into the Arduino by clicking the Upload button.

The sketch starts automatically. Assuming the motors have been connected properly, the motors should turn in various directions as the robot goes through its motion routines.

Fully test the robot by first disconnecting the main power to the main battery pack. Plug the nine volt battery into the Arduino, and reconnect the main power. Place the robot on flat ground; the robot should go through its motion routines: forward, backward, turn right, turn left, and stop.

If one or both motors turn in the wrong direction, remove power and flip the terminal wiring from the affected motor on the H-bridge.

Sources

Precut and predrilled Beginner Bot base, with all construction hardware:

Budget Robotics
www.budgetrobotics.com

Arduino, prototyping shield, Seeedstudio L298 motor bridge module (see Part 2 of this series for details), mini solderless breadboards, jumper wires, header pins, etc.:

Adafruit Industries
www.adafruit.com

All Electronics
www.allelectronics.com

Electronix Express
www.elexp.com

HVW Tech
www.hvwtech.com

Jameco Electronics
www.jameco.com

Mouser Electronics
www.mouser.com

Pololu
www.pololu.com

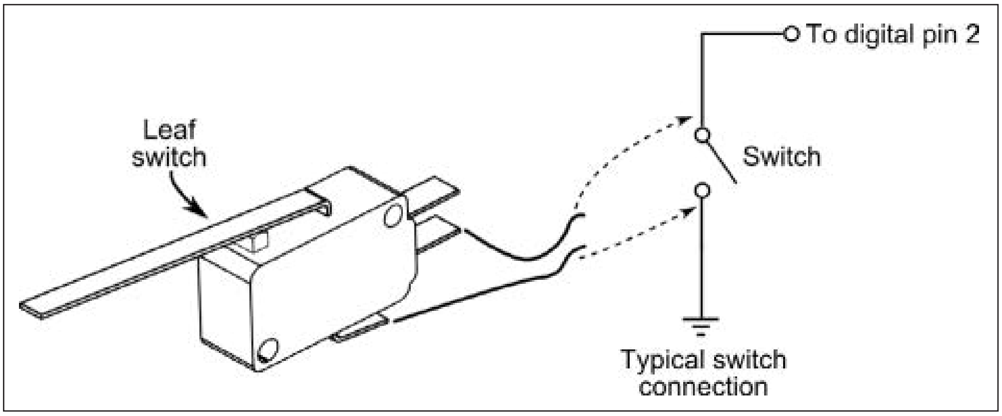
RobotShop
www.robotshop.com

Schmartboard
www.schmartboard.com

SparkFun
www.sparkfun.com

Solarbotics
www.solarbotics.com

FIGURE 5. A spring-loaded leaf switch makes a good object contact bumper. When the switch closes, the Arduino can register it as a collision with something.



Ground clearance of the Beginner Bot is limited, so the best surfaces for testing include tile, wood, or a kitchen table. Carpet is acceptable as long as it has a very low nap. If the bot appears to struggle as it's moving along, relocate it to ride over a smoother surface.

Flashlight, Flashlight, Shining Bright

Both Parts 2 and 3 of this series showed how to control the Beginner Bot using a flashlight by shining the light into the photocell eyes. **Listing 2** provides the same method, this time with an Arduino. The sketch tells the Arduino to read the value from both photocells. A series of If conditional logic tests determine if there's enough light to follow, and if so, in what direction the robot should travel.

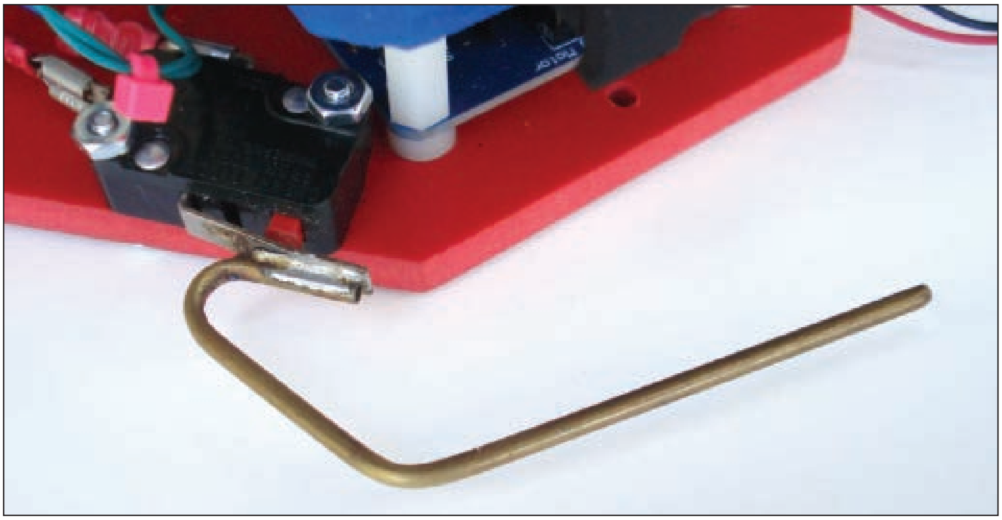


FIGURE 6. Mount the switch on one side of the front of the Beginner Bot. By attaching a stiff wire to the leaf (solder or glue), you can extend its reach across the entire front of the robot.

The program first sets a threshold value to determine the boundary between dark and light, as well as determining the ambient (natural) level of light in the room. I've set the light/dark threshold to 800 (out of a range of 0-1,023) as a starting point. Try higher or lower values to see what works best with your particular CdS cells. I've set the ambient level at 600.

- When both cells receive light over the ambient level, the robot drives forward.
- When only one cell receives light over the threshold, the robot turns in the direction of the light.
- If neither cell receives light over the ambient level, the robot stops.

Upload the program in **Listing 2**. When transfer is complete, remove the programming cable. Move to a darkened room, apply power to the robot, and place it on the ground. Aim a bright flashlight away from the Beginner Bot. The robot should not move. Next, shine the flashlight evenly into both photocells. The robot should move toward you. Get close to the robot and aim the flashlight into just one photocell. The robot should turn toward the photocell with the light shining into it.

If your robot moves when there's no light falling on the CdS cells, try changing the ambient and threshold values. Conversely, if the light from the flashlight seems to make

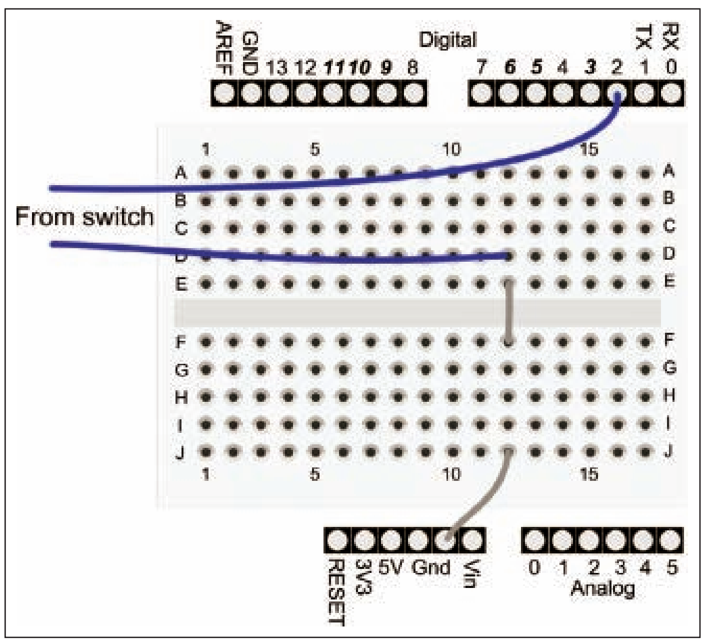


FIGURE 7. Breadboard view of attaching the wires from the switch to the Arduino. Connect one lead from the switch to ground; the other to digital pin D2.

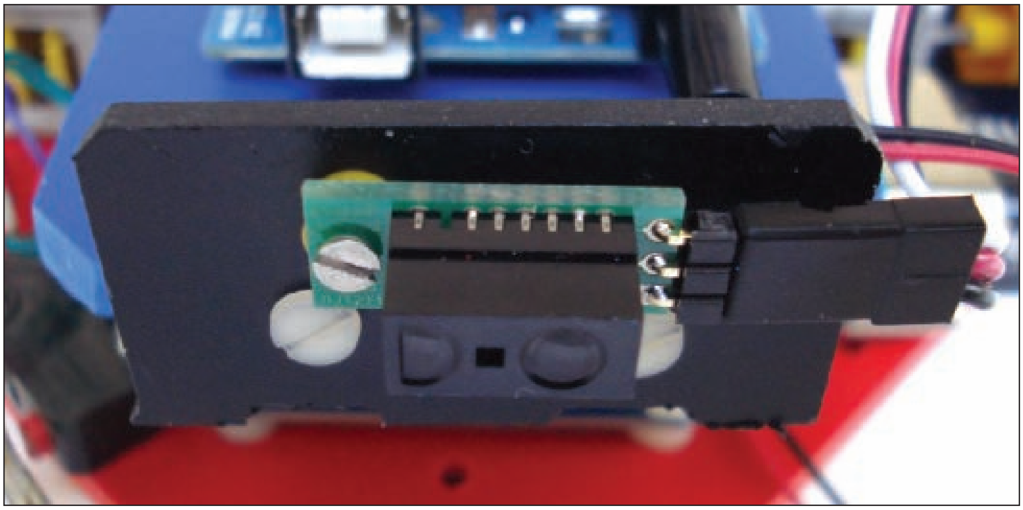


FIGURE 8. The GP2Y0D810 infrared distance judgment sensor, mounted on the front of the Beginner Bot on a homemade bracket. I'm using an '810 module attached to a breakout board.

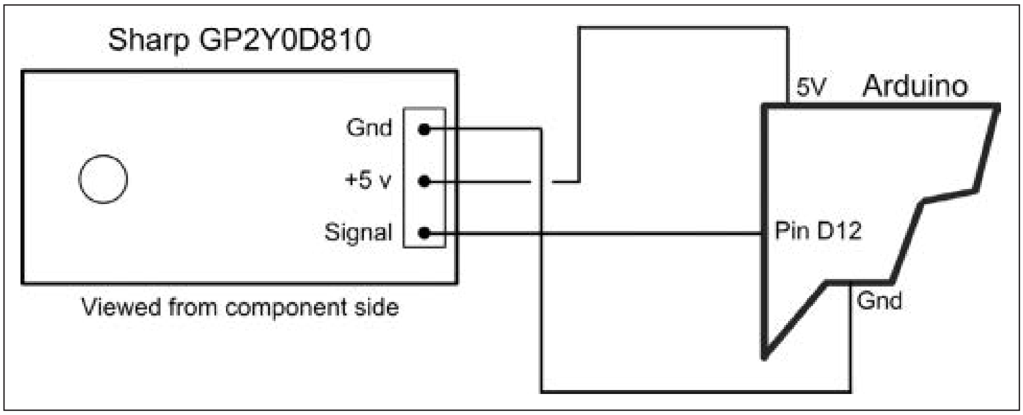


FIGURE 9. Wiring diagram for connecting the GP2Y0D810 to the Arduino. Be sure to observe the correct polarity of the 5V and ground connections. (See the silkscreen printing on the sensor breakout board itself.)

no difference, enter a lower threshold and try a darker room.

Adding Object Detection Sensors

With one or two simple and affordable sensors, you can extend the usefulness of the Beginner Bot to detect objects in front of it, and move out of the way if something is there.

Perhaps the simplest method of object detection is the good ol' bumper switch, shown in **Figure 5**. The switch is connected between ground and digital pin 2 of the Arduino. When the switch closes, the circuit is completed, and the Arduino registers it as a contact with an object.

I like to use leaf switches as bumpers because they already have a fairly large area of contact. You may even wish to extend the bumper of the switch by soldering or gluing a piece of coat hanger wire to the leaf, like that in **Figure 6**. I've bent the wire to form a bumper zone that extends across the center of the Beginner Bot.

Figure 7 shows how to connect the switch to the Arduino. One easy method is to use 22 gauge solid conductor wire, and merely string wires from the connection points on the switch to the breadboard and digital pin D2 on the Arduino. Most leaf switches use male quick disconnects; rather than soldering directly onto these, buy two female crimp-on connector tabs. Strip extra off the end of the wire and fold it over to double its thickness. Then, use a crimping tool or large pliers to secure the connection.

Wiring to the Arduino is simple and takes advantage of the Arduino's built-in pull-up resistors. These resistors — which are activated in software — ensure that when the switch is open, its output is a consistent and reliable HIGH value. When the switch is closed, the output goes LOW.

Another relatively inexpensive sensor is the digital infrared proximity detector. I've selected the Sharp

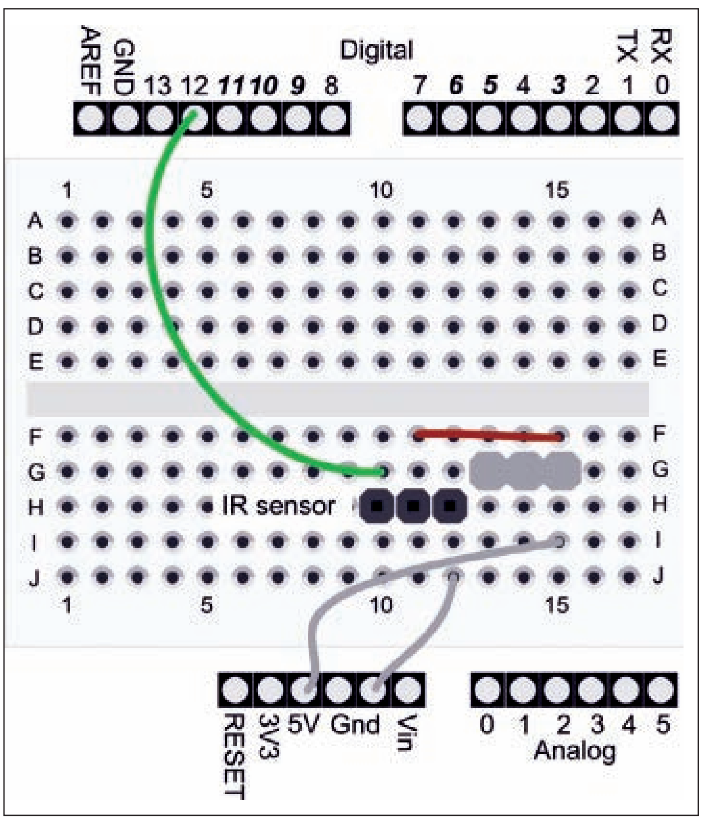


FIGURE 10. Breadboard view of the GP2Y0D810. Use a standard three-wire servo extension (6" to 8" is enough) to connect the sensor to the breadboard.

GP2Y0D810. It's available from Pololu — among other sources — and is priced at about $7. The board has a single mounting hole and comes with a three-pin header that you need to solder into place. Attach a three-wire servo extension cable between the sensor and the Beginner Bot's breadboard. (Be sure to get some extra double-length male headers while you're at it — Pololu has these, as well. You use them to attach the female end of the servo extension to the solderless breadboard.)

I've connected the sensor for my Beginner Bot to a homebrew mounting bracket. Because the bracket is fairly wide, I made it so that it could be swiveled out of the way when I need to plug the USB cable into the Arduino.

The mounted IR detector is shown in **Figure 8**. Refer to the wiring diagram in **Figure 9** for the basic connection, and the breadboard view in **Figure 10**.

See **Listing 3** for a demonstration of using the Beginner Bot with the switch and infrared object detection sensors. The sketch first reads the current output of the switch. If it's LOW, the switch has been triggered. The robot reacts by momentarily backing up, turning to the right, and then proceeding forward again.

Next comes a test to see if the IR detector has been activated. If it's LOW, there's an object within its field of view which extends out to about 10 centimeters in front of the sensor. If triggered, the robot backs up briefly, turns to the left, and then continues.

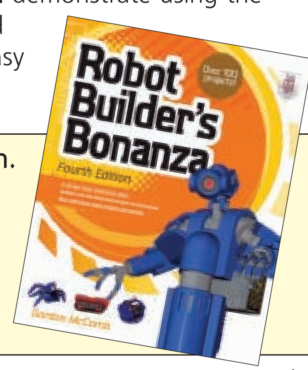
LISTING 3

```
const int waitDelay = 2000;     // Delay between motion routines
const int in1 = 7;              // Input1 pins
const int in2 = 8;
const int in3 = A1;             // Input 2 pins
const int in4 = A0;
const int bumper = 2;           // Right bumper pin 2
const int irSense = 12;         // IR sensor pin 12

const int ledPin = 13;
int pb = 0;
int ir = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  pinMode(bumper, INPUT);       // Set to input
  digitalWrite(bumper, HIGH);   // Turn on pullup resistor

  // Indicate proper operation
  digitalWrite(ledPin, HIGH);
  delay(500);
  digitalWrite(ledPin, LOW);
  delay(500);
  digitalWrite(ledPin, HIGH);
}

void loop() {

  goFwd();

  // Test bumper switch
  pb = digitalRead(bumper);
  if (pb == LOW) {              // If bumper hit
    goRev();
    delay(500);
    goRight();
    delay(1500);
  }

  // Test
  ir = digitalRead(irSense);
  if (ir == LOW) {              // If IR triggered
    goRev();
    delay(500);
    goLeft();
    delay(1500);
  }

  delay(10);

}

[duplicate "motion routines" from Listing 1 here]
```

Finishing Up: Adapting the Beginner Bot to the Parallax Propeller

The Beginner Bot is microcontroller agnostic; it doesn't mind how it's controlled or what's controlling it. So far, we've examined two popular microcontrollers, and next month we'll conclude with interfacing the Beginner Bot to a Parallax Propeller.

The "Prop" offers numerous features not found in many other microcontrollers. I'll demonstrate using the Beginner Bot with an enhanced Propeller board designed for easy development. **SV**



Gordon McComb is the author of *Robot Builder's Bonanza*, now in its fourth edition. Greatly expanded and updated, this best selling book covers the latest trends in amateur robotics, and comes with 10 all new robot construction projects, plus more ideas for building robots from found parts. Look for *Robot Builder's Bonanza, 4th Ed* in the *SERVO* Webstore at **http://store.servomagazine.com**. Gordon may be reached at rbb@robotoid.com.

The *SERVO* Webstore

<u>Attention</u> Subscribers ask about your discount on prices marked with an *

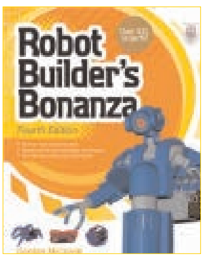
CD-ROM SPECIALS

ROBOTICS

Robot Builder's Bonanza, Fourth Edition

NEW!

by Gordon McComb

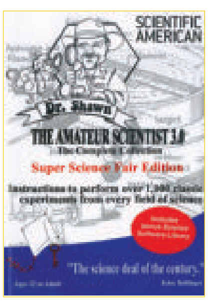


Robot Builder's Bonanza, Fourth Edition includes step-by-step plans for a number of motorized platforms. The book is described as a compendium of robotics topics, containing more than 100 projects, including 10 robot designs new to the fourth edition. These modular robots are low cost, and are made to be reproduced by readers with no training in mechanical construction.

$29.95*

The Amateur Scientist 4.0 The Complete Collection

by Bright Science, LLC



There are 1,000 projects on this **CD**, not to mention the additional technical info and bonus features. It doesn't matter if you're a complete novice looking to do your first science fair project or a super tech-head gadget freak; there are enough projects on the single CD-ROM to keep you and 50 of your friends busy for a lifetime!

Reg $26.95 Sale Price $23.95

Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists

by Dustyn Roberts

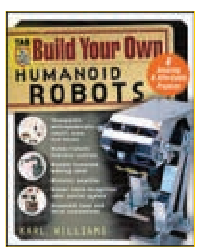
In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.

$29.95*

Build Your Own Humanoid Robots

by Karl Williams

GREAT 'DROIDS, INDEED!

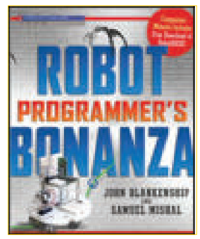


This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot.

$24.95*

Robot Programmer's Bonanza

by John Blankenship, Samuel Mishal



The first hands-on programming guide for today's robot hobbyist!

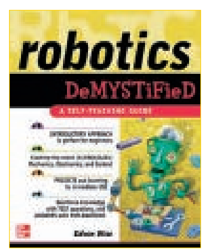
Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.

$29.95

Robotics Demystified

by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!

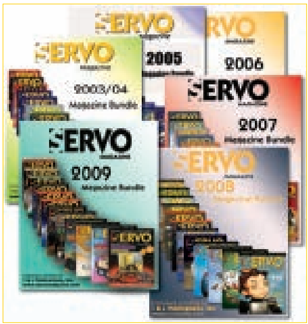


Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

To order call 1-800-783-4624

SERVO Magazine Bundles



Published by T & L Publications, Inc.

$57 per bundle

Save $10 off the normal price!!

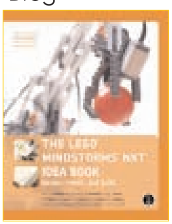
Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, and 09.

Call my Webstore and you'll get someone in AMERICA!

Visit my online store @ www.servomagazine.com

LEGO MINDSTORMS NXT Idea Book

by the Contributors to The NXT Step Blog

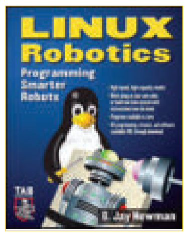


If you're serious about having fun with LEGO® robotics, you've come to the right place. The team behind The NXT STEP blog — the authoritative online source for MINDSTORMS® NXT information and advice — has packaged its considerable skills and experience in this book. Inside, you'll find some of the team's best ideas for creating cool and sophisticated models, including instructions for eight robots you can build yourself.

Reg $29.95 Sale Price $24.95

Linux Robotics

by D. Jay Newman



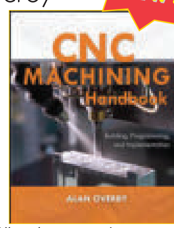
If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.

$34.95

CNC Machining Handbook: Building, Programming, and Implementation

by Alan Overby

NEW!

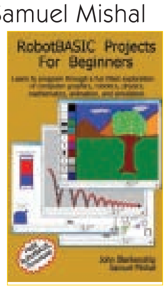


The *CNC Machining Handbook* describes the steps involved in building a CNC machine and successfully implementing it in a real world application. Helpful photos and illustrations are featured throughout. Whether you're a student, hobbyist, or business owner looking to move from a manual manufacturing process to the accuracy and repeatability of what CNC has to offer, you'll benefit from the in-depth information in this comprehensive resource. **$34.95**

SPECIAL OFFERS

RobotBASIC Projects For Beginners

by John Blankenship, Samuel Mishal



If you want to learn how to program, this is the book for you. Most texts on programming offer dry, boring examples that are difficult to follow. In this book, a wide variety of interesting and relevant subjects are explored using a problem-solving methodology that develops logical thinking skills while making learning fun. RobotBASIC is an easy-to-use computer language available for any Windows-based PC and is used throughout the text.

Reg. Price $14.95 Sale Price $9.95

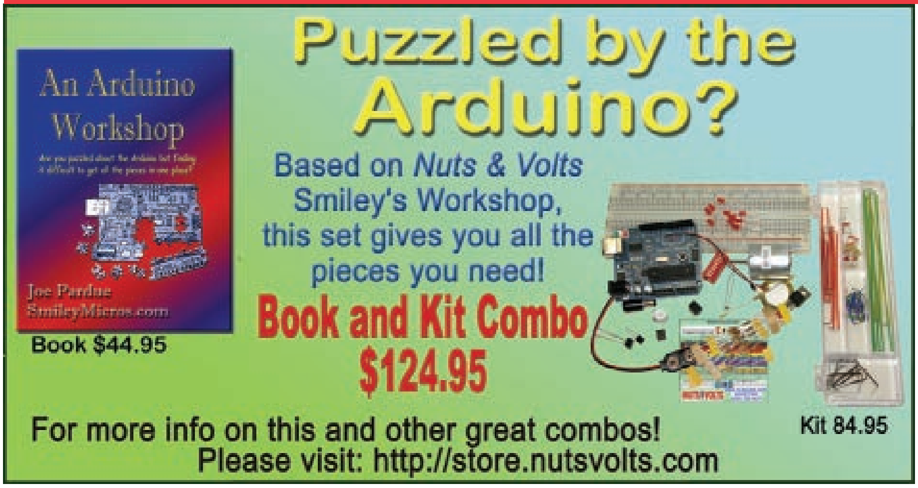
Technology Education Package for Everyone Starting in Electronics

This lab – from the good people at GSS Tech Ed – will show you 40 of the most simple and interesting experiments and lessons you have ever seen on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit. Along with the purchase of the lab, you will receive a special password to access the fantastic online interactive software to help you fully understand all the electronic principles. For a complete product description and sample software, please visit our webstore.

Regular Price $79.95 **Subscriber's Price $75.95**

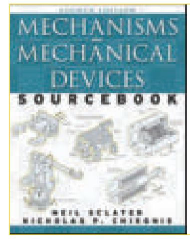
Or order online www.servomagazine.com

SPECIAL OFFERS



Mechanisms and Mechanical Devices Sourcebook

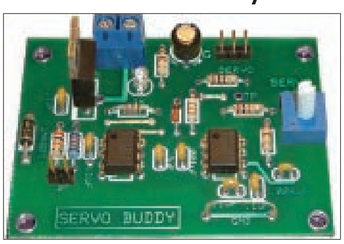
by Neil Sclater, Nicholas Chironis



Over 2,000 drawings make this sourcebook a gold mine of information for learning and innovating in mechanical design. Overviews of robotics, rapid prototyping, MEMS, and nanotechnology will get you up to speed on these cutting-edge technologies. Easy-to-read tutorial chapters on the basics of mechanisms and motion control will introduce those subjects to you. **Reg $89.95 Sale Price $69.95**

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.

For more information, please check out the May 2008 issue or go to the SERVO webstore.

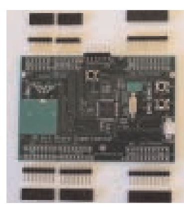
Includes an article reprint.

Subscriber's Price **$39.55**

Non-Subscriber's Price **$43.95**

32-Bit Micro Experimenter Board

The new 32-Bit Micro Experimenter is the fastest way to learn 32-bit microcontrollers.



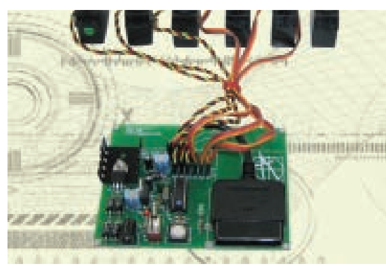
NEW!

The kit includes onboard 46 programmable I/O and USB, free software, carefully documented step-by-step experiments for USB, embedded web server, graphics and audio, wireless, RTOS, and file I/O. User pushbuttons, LEDs, and 32 kHz clock crystal. Can be used in solderless breadboard environment or stand-alone. Also supports Arduino compatible interface.

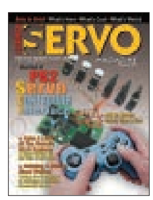
Subscriber's Price **$89.95**

Non-Subscriber's Price **$93.95**

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors. Includes all components and instruction manual.



For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

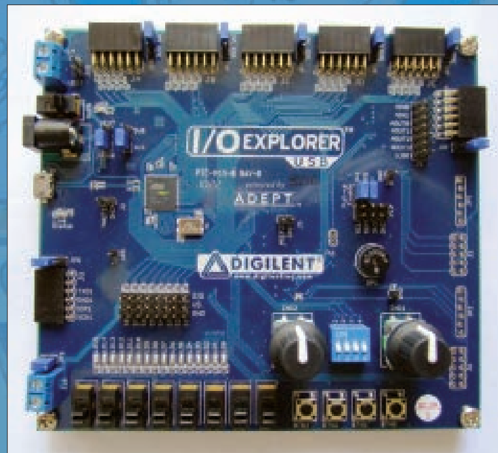
Subscriber's Price **$79.95**

Non-Subscriber's Price **$84.95**

Twin Tweaks

Bryce Woolley & Evan Woolley

THIS MONTH: Due Digilence



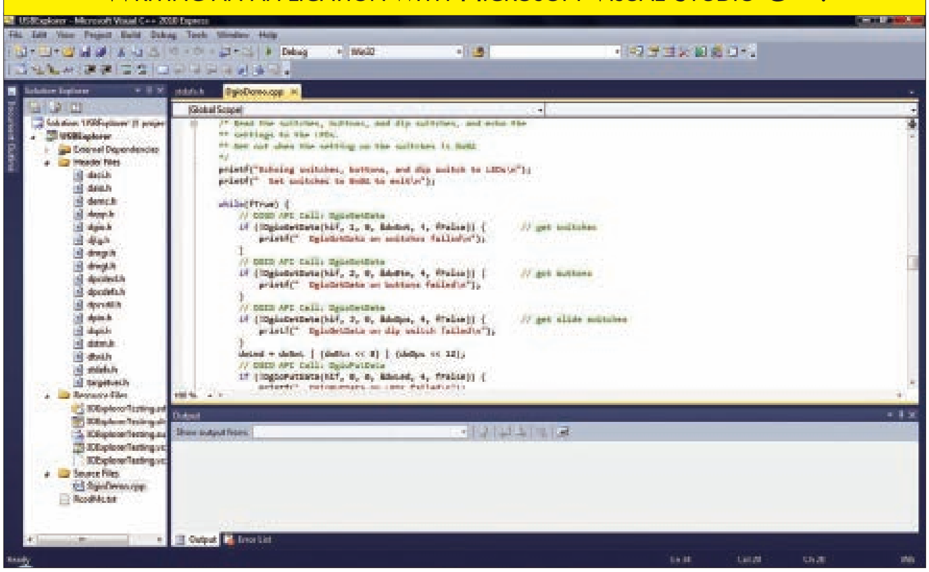
THE USB I/O EXPLORER FROM DIGILENT.

This month, we have the opportunity to present the USB I/O Explorer from Digilent. The I/O Explorer is a dedicated USB peripheral device choc full of sensors so intrepid roboticists can practice their programming skills by writing applications to interact with real world inputs. The I/O Explorer is a bit different from the projects we're used to – with Evan's mechanical engineering background, we're most accustomed to adding new mechanisms and sensors to tweak kits. We would describe our programming skill set as instrumental – we do what it takes to get by, kind of like MacGyver. We shoehorn lines of code together that might not normally have any business next to each other, but it will get the robot through the maze in one piece. The I/O Explorer, however, is all about the coding. The sophisticated array of sensors and expansion ports take the guesswork out of equipping your device with the means to gather data from the outside world, allowing you to focus on writing the code capable of dealing with such a wealth of information. We hope the I/O Explorer will help us refine our coding skills to be a little more James Bond and a little less MacGyver.

I/O Explorer: First Class

The I/O Explorer is an impressive and highly evolved device, so it makes sense that the process of designing it was one of experimentation and incremental advancement. Before the I/O Explorer burst onto the scene, the diligent folks at Digilent were working on a number of Field Programmable Gate Array (FPGA) boards. The Adept software system was originally designed to download configuration files to the FPGA boards. The Adept software was soon expanded to move data back and forth between the FPGA boards and the software application on the PC. The culminating result is the Adept software application that allows users to discover connected devices, configure them, and transfer data between the device and the computer. The software application has

WRITING AN APPLICATION WITH MICROSOFT VISUAL STUDIO C++.



an intuitive tabbed interface that logically groups the different operations that you would want to perform with the device.

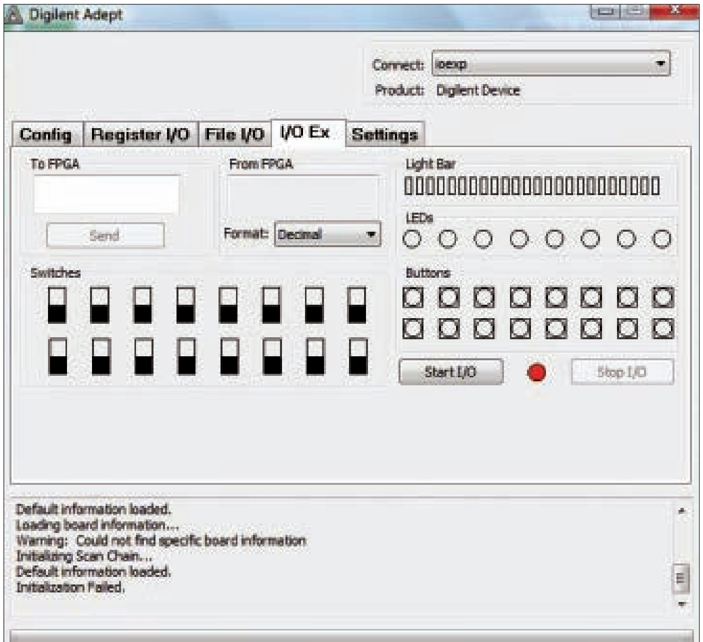
Firing up the application, we were happy to see that it recognized our I/O Explorer upon connecting it to the computer via the provided retractable USB cable. But beyond seeing that an "ioexp" was connected, we really couldn't do anything with the Adept software. The board was never able to initialize while using the Adept application. We were stunned that we were running into problems so soon, but then we did what any wise tinkerer would do in the first place and took a penetrating look at the reference manual (which is available on the Digilent website).

The Adept software application is not designed to work with the I/O Explorer. The I/O Explorer is based on AVR microcontrollers, the in-system programming for which is SPI based. This is not a cause for alarm, because the folks at Digilent have created a vast library of API sets to help users take full advantage of the impressive capabilities of the I/O Explorer. Instead of using the Adept application, users must simply write applications that run on the computer and use the I/O Explorer as an input device. Before we got too deep into programming, we wanted to take a moment to review the impressive array of sensors and other inputs on the board.

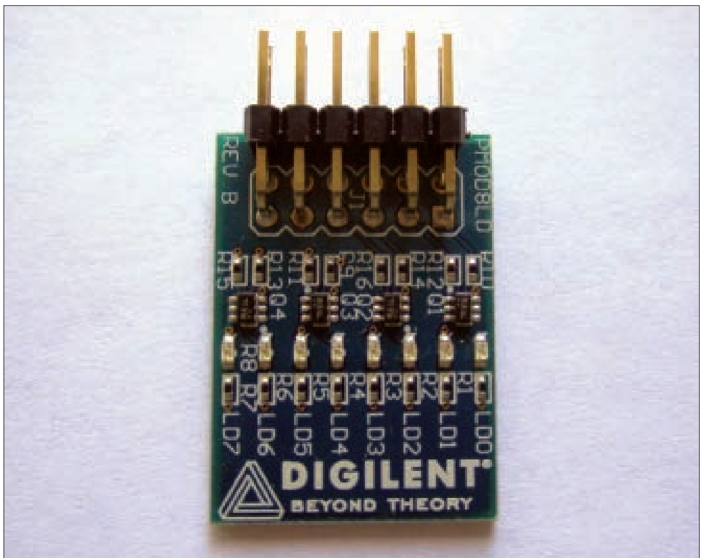
The I/O Explorer itself is a sleek and polished unit, with a plethora of components sitting atop a cool blue PCB. Immediately apparent are sets of switches and LEDs. A row of 16 LEDs sits above a row of eight toggle switches. Next to the switches are a set of four pushbuttons. In addition to the tactile, the I/O Explorer covers your auditory senses by including a buzzer near the center of the board. Rising up from the board like majestic skyscrapers above a PCB city are two encoder knobs with built-in pushbuttons. There are two 10-bit analog-to-digital inputs and four 12-bit digital-to-analog outputs. The board even contains eight ports for connecting RC servos. Finally, lining the edge of the board are six Pmod connectors (five 12-pin and one six-pin). The Pmods are peripheral modules that add functionality to the board, but before looking at those there was still plenty to do with the stock device.

Are You Smarter Than a PhD Candidate in Computer Science?

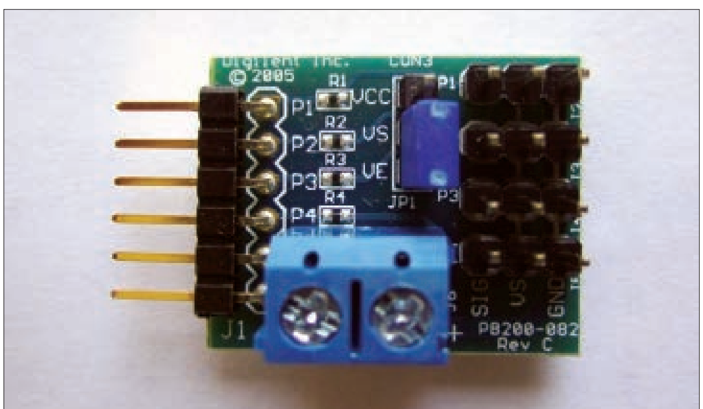
Even though the I/O Explorer does not work with the Adept software application, it is compatible with the Adept Software Development Kit available on the Digilent website. The Adept SDK is a bundle of sample codes and documentation that offer plenty of guidance accessible even to instrumental programmers like ourselves. That being said, the I/O Explorer is not exactly designed for programming novices. We had a difficult time ourselves getting started with the device, but that was partly from a misunderstanding engendered by a far too cursory reading of the reference manual. We are used to projects with



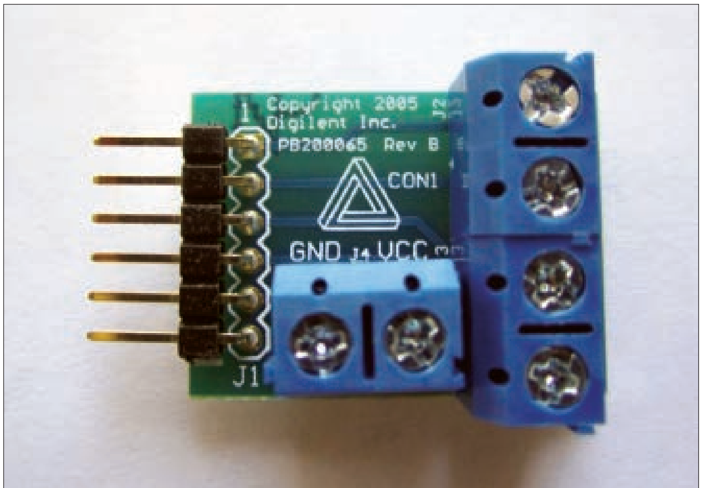
THE ADEPT INTERFACE.



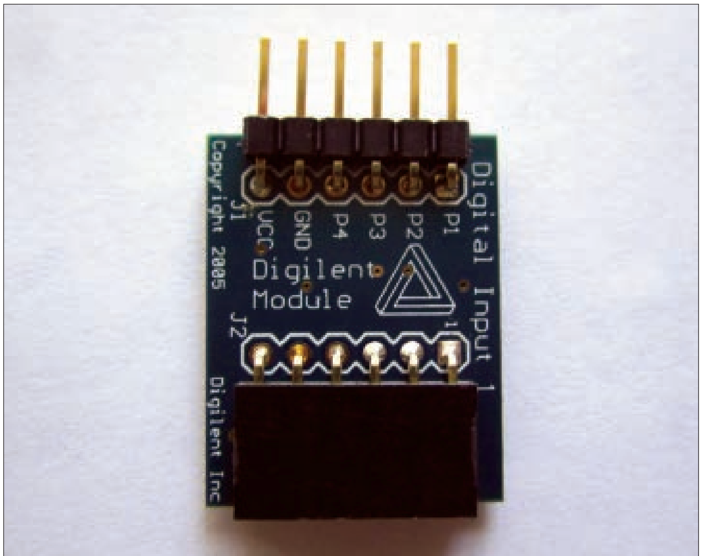
EIGHT LED PMOD.



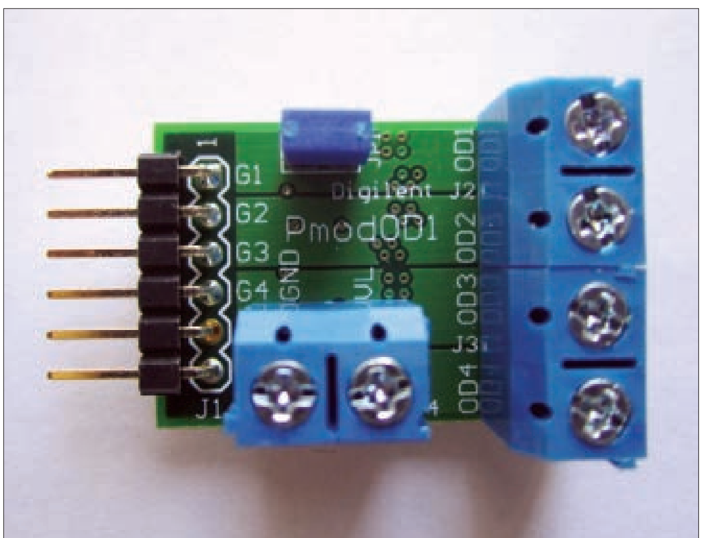
THE CON3 PMOD FOR ATTACHING SERVOS.



THE CON1 PMOD COMES WITH HANDY WIRE TERMINALS.



A PMOD FOR HANDLING DIGITAL INPUTS.



PMOD OD1, WHICH CAN BE USED TO SUPPLY PLENTY OF POWER TO MOTORS.

robotics kits, where programs are downloaded to and run on the robot itself. At first blush, we hoped the I/O Explorer would be the same way – it had not one, but two AVR microcontrollers and we thought we could engage in our familiar routine of downloading programs to the board and running them on the unit. But alas, the primary function of the I/O Explorer was to act as a peripheral device, simply providing input to programs running on the computer. This doesn't sound like an insurmountable task, and we didn't think it would be – we just didn't know where to get started.

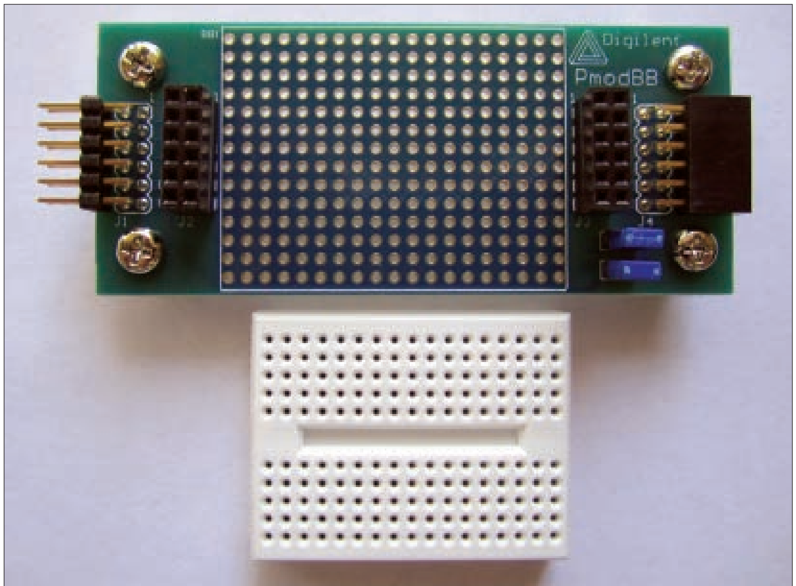
Initially, we thought we would be able to use one of the environments with which we were familiar. Microchip's MPLAB, however, is meant to be used with Microchip microcontrollers. With that in mind, we looked to AVR Studio given the I/O Explorer's use of the AT90USB646 and ATmega165P Atmel microcontrollers. This didn't seem to work either, because AVR Studio sometimes needs a more direct connection to the Atmel microcontroller, usually by something like the AVR In-Circuit Emulator. Whatever the case, AVR Studio didn't seem to recognize the Atmel microcontrollers embedded in the I/O Explorer. Figuring that Micro Code Studio and Programmer's Sketchpad would be similarly fruitless, we mustered up the courage to foray into new IDE (Integrated Development Environment) territory.

The most challenging part of getting started with the I/O Explorer was getting acquainted with a new IDE. While the files included in the SDK can ostensibly be implemented by any number of IDEs, we eventually decided to go with the one that was used for the initial development of the applications by the folks at Digilent – Microsoft Visual Studio. We're more comfortable with the well appointed interface of MPLAB or the Spartan appeal of WinAVR – the lavish and labyrinthine Microsoft Visual Studio was initially as disorienting as Kevin Flynn's first trip to the Grid. We suppose that disorientation stems from the vastness of Visual Studio when compared to some of the other IDEs we've used. Instead of simply writing source files that are downloaded directly to a robot, Visual Studio allows users to create sophisticated applications that bring together a veritable army of different files and file types.

The Adept SDK comes with a veritable army of files, and we would have to be a little more crafty than opening the C++ source file and running it. Thankfully, Visual Studio does have a wizard for starting a console application that can be run from the command line which is exactly what you need to do to interact with the I/O Explorer. With the wizard ready to cast its spell, we just needed a sample program to get started with.

The samples included in the Adept SDK help show off every capability of the I/O Explorer and its Pmods. Every sample includes a C++ source file, all of the necessary header files, and a short text file describing what the sample program does. Even though the source files are C++, the code is completely compatible with C because it

doesn't use any functions peculiar to C++. There is even a reference manual for each sample that describes all of the functions and syntax used in that project. Most of the kits that we've worked with before usually include only one reference manual for the entire kit. This can result in voluminous guides that require a bit of searching through to find the exact function you're curious about. The numerous but specific guides for the Adept SDK seemed a lot more helpful, particularly for a kit where a single volume reference manual would perhaps rival *War and Peace*.



A BREADBOARD PMOD PERFECT FOR RAPID PROTOTYPING.

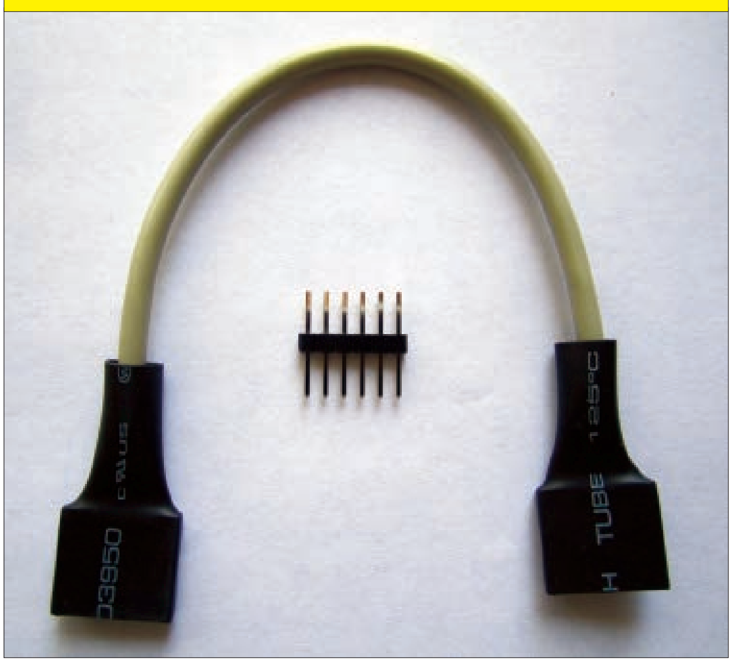
After browsing through the samples, we thought that the DGIO sample program would be a perfect place to start. The DGIO program is a way for the user to interact with the I/O Explorer via the switches, pushbuttons, and LEDs. It seemed like a great way to get acquainted with the Explorer, and the perfect basis for our first ever Visual Studio application. Microsoft Visual Studio, however, is not the kindest IDE to outsiders.

For a while at the beginning, getting the sample programs running was as mind bending as the Penrose Triangle that serves as Digilent's logo. Even with the wizard, we still had some issues with getting all of the proper files in the proper places. It might sound a bit silly, but there is quite a steep learning curve for getting started with Microsoft Visual Studio. Unfortunately, the otherwise excellent documentation from the Digilent folks is of no help here – they understandably did not want to presuppose what IDE people would be using. We appreciate such an effort, but a quick start guide about how to create suitable applications for interacting with the I/O Explorer would have been nice too.

Perhaps we're being too needy. Of course, it would have made it easier on us to have a guide to walk us through application creation with kid gloves, but Visual Studio itself has numerous wizards and voluminous help files. We figured out the process eventually, and we chalk up our slow going to the fact of being complete Visual Studio neophytes. Once again, the I/O Explorer is not exactly meant for neophytes. While a programming beginner could conceivably cut their teeth with the I/O Explorer, those already fluent will get the most out of the product. Even if our theoretical novice was unfazed by the labyrinth of Visual Studio, the I/O Explorer — even with the delightfully straightforward DGIO program — is no replacement for Hello World.

The I/O Explorer, however, is perfect for the more confident programmers looking for a way to hone their skills. Sophisticated programmers can make everything from Rubix cube solvers to protein sequencers, but a limitation on many sophisticated programs is that they are run in the abstract (to the extent that staying within the confines of the computer is abstract). On the other hand of the concreteness spectrum are many robotics kits. One of the most exciting things to us about programming robotics kits is the chance to see lines of code become real world behaviors. Many robotics kits, however, are limited in the number of add-ons and extra sensors you can include. These are understandable limitations considering that robot kits need to accommodate a means for locomotion and other sometimes bulky mechanisms. The I/O Explorer, however, inhabits the Goldilocks zone by providing a platform to bring your fancy code into the real world that's as expandable and versatile as your programming skills can handle. We just scratched the surface with the simple DGIO program, but once you get comfortable with Visual Studio it truly opens the floodgates of possibility.

A SIX-PIN CABLE FOR CONNECTING YOUR OWN CUSTOM PMODS.



THE 2X6 PIN CABLE TO DUAL SIX-PIN CABLE.

Pmod Squad

One of the best ways for an adventurous programmer to test their coding chops is with the menagerie of Pmods available from Digilent. Pmods are Digilent's brand of peripheral modules. The Pmods are compact I/O interface boards meant to add functionality to the I/O Explorer and other Digilent boards. We had the opportunity to test out a delightful variety of Pmods, many of which could be implemented by sample programs in the SDK.

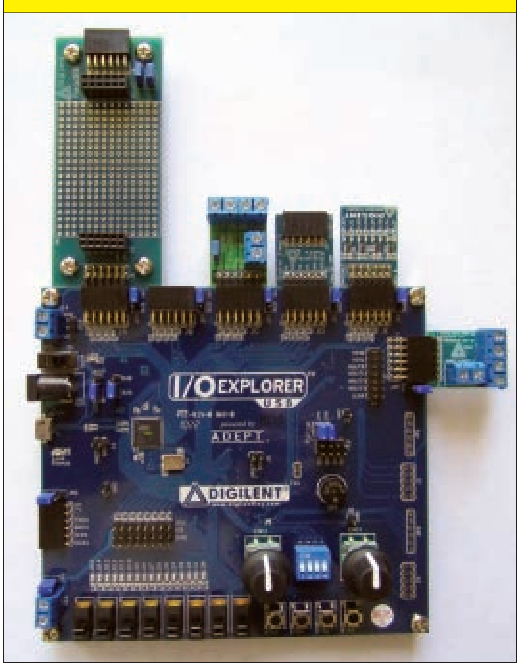
One of the most basic Pmods is the eight LED Pmod. The simple unit contains exactly what you would expect – eight small LEDs arranged in a row. Turning LEDs on and off can make for an amusing light show, but what they're really useful for is debugging. Coordinating LED flashes with input from other sensors is a great way to see that your sensors are working properly. Sure, you can add printed messages to the output window of your application, but flashing LEDs are an exciting and visual representation of your sensor inputs. The eight LED Pmod even has its own sample program included in the Adept SDK.

For those hackers longing for a mechanism, Pmods provide even more opportunities to add servos and motors. Even though the I/O Explorer comes with eight ports for RC servos, the CON 3 Pmod comes with four more ports. Running 12 servos sounds like a tall order, so the CON3 Pmod comes with wire terminals so an external power source can be connected. Speaking of power hogs, the OD1 (open drain output) Pmod allows you to power high current devices by sourcing up to 20V and 3.0A continuously. The CON1 Pmod can be used for more conservative devices. It includes two terminals for external power, and four terminals can be used for data signals.

Another sleek Pmod is the digital input Pmod which is a great way to implement whatever new and exciting sensor the user's heart desires. The digital input has a resolution of 12 bits, and the Pmod comes with a six inch six-pin cable and header. We think the cable is a welcome addition because without it, the I/O Explorer board can get a bit crowded when using a few Pmods at once.

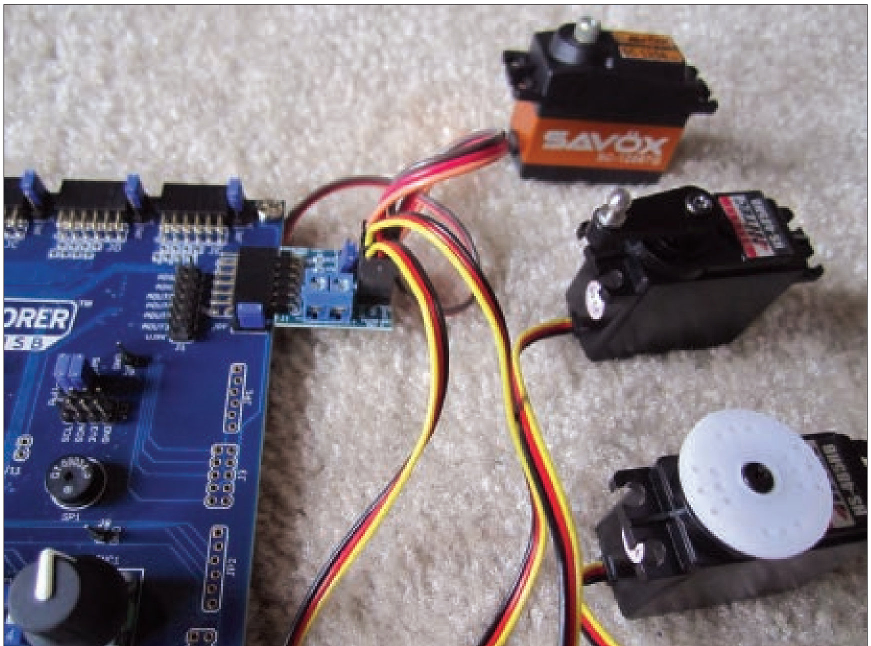
Another Pmod that we find particularly exciting is the breadboard Pmod. Larger than the others, this Pmod supports a small breadboard where users can construct their own circuits. While a breadboard is a great way to prototype new circuits, no matter how nicely you arrange your components on a breadboard, the circuit usually

PMODS GALORE!



PROGRAMMING APPLICATIONS.

retains the sense of being unfinished or unpolished. The I/O Explorer comes with the means for intrepid hackers to make their own custom Pmods by providing the schematics for the I/O Explorer and all of the Pmods. The extra cables and headers would also make implementing custom Pmods a bit easier by taking off the pressure of recreating the Pmods' minimalistic use of real estate.



SERVO PARTY!

The Expandability Never Dies

The wildest thing is that even with all of the Pmods we had the chance to preview, we only scratched the surface of the wide selection from Digilent. It's as if the folks at Digilent have created an electronics version of Willy Wonka's chocolate factory – everything you could imagine and some things you couldn't are available to interface with the I/O Explorer. There are Pmods with even more super bright LEDs, additional switches, and more pushbuttons. There's a Pmod with the RS-232 port that we so despise (we were overjoyed that the Explorer is indeed a USB I/O Explorer). There are three-axis gyroscopes and accelerometers (more perfect candidates for use with an extra cable, which both come with). There are infrared light sensors and even thermometers. There are RCA audio jack Pmods and a Pmod that can connect to a computer mouse. And just in case all of these different Pmods sound like they might lead to a lot of data gathering, there are Pmods for using an SD card for storage.

The Pmods are delightfully affordable, with many starting as low as $9.99. There are, of course, fancier Pmods that come with a fancier price tag, including the 802.11b/g/n wireless interface Pmod at $59.99. The wide variety of Pmods correspond to a number of sample programs in the SDK, but the most exciting thing about the Pmods is that they provide excellent inspiration for what the I/O Explorer is really meant for – users writing their own custom applications.

The USB I/O Explorer can be used as a programmable microcontroller, so if you ever wanted to outfit your robot with a brain that is extraordinary instead of abby-normal, your search is over. To be used as a brain for a robotics kit, the only major physical modification that needs to be made is to equip the Explorer with its own power source. Instructions to do just that are included with the board's documentation, and the number of extra parts needed is minimal. The board comes with three connectors that can be used to connect an external power supply, and the Digilent website offers battery holders that can be connected with ease.

Using the I/O Explorer as a robot brain involves writing custom firmware, but the original firmware is available on the Digilent website so hackers need not worry about crossing a coding Rubicon. Such an endeavor might be perfect for a future project that would be a little more familiar to our roboticist predispositions.

Even with our predispositions, the I/O Explorer provides a valuable tool for roboticists aspiring to be well rounded. With a mechanical engineering background, Evan is particularly focused on things like end effectors and clean circuitry when it comes to a robotics project, but in many cases a robot can only be as good as its program. The software is what really shapes how a robot reacts to the world, and if your code can't make sense of the input from sensors, then your bot is about as useful as the Beagle II.

The I/O Explorer is a challenging device, but its broad capabilities inspire us to be better programmers. With a smorgasbord of sensors at our fingertips and a peripheral device with seemingly endless expandability, we no longer have an excuse not to be! Despite the initial bumps in the road because of our Visual Studio ineptitude, we had a great time exercising our programming muscles, and the preliminary problems simply made our eventual victory all the sweeter. Of course, we know it's too soon to truly call it a victory because programming is a bit like the game of chess – easy to pick up and difficult to master. The USB I/O Explorer has the capability even to challenge the most masterful of programmers. **SV**

RECOMMENDED WEBSITES

www.digilentinc.com

SPECIAL THANKS TO

Gene Apperson and Alex Wong

Continued from page 21

patiently for NASA to explore."

The innovative advanced suspension systems offered in the Super Rover Kit are ideal for navigation of the chaotic outdoor environment. The chassis provides ample space and carrying capacity to incorporate microcontrollers, sensors, cameras, and other electronics needed to effectively compete.

"Historically, the complexity and costs associated with AUV competitions have hindered growth and participation," explains Marzetta. "MINDS-i is pioneering a sleeker and more robust system at a lower cost to help promote the expansion of this important category of relevant robot competitions."

For further information, please contact:

MINDS-i, Inc.
22819 E. Appleway Ave.
Liberty Lake, WA 99019
Tel: **509•252•5767**
Email: **info@mymindsi.com**
Website: **www.mymindsi.com**

Laser Rangefinder

Designed in conjunction with Grand Idea Studio, the Laser Range Finder (LRF) module available now from Parallax is a distance-measuring instrument that uses laser technology to calculate the distance to a targeted object. The design uses a Propeller processor, CMOS camera, and laser diode to create a low cost laser range finder. Distance to a targeted object is calculated by optical triangulation using simple trigonometry between the centroid of laser light, camera, and object. Features include:

- Compact module with integrated CMOS camera and laser system.
- Optimal measurement range of 6–48 inches (15–122 cm) with an accuracy error <5%; average 3%.
- Maximum object detection distance of ~8 feet (2.4 meters).
- Range finding sample rate of 1 Hz.
- Single row, four-pin, 0.1" header for easy connection to a host system.

For further information, please contact:

Parallax, Inc.
Website: **www.Parallax.com**

New Economical 5/50 MHz Signal Generators

Saelig Company, Inc., has introduced two new, budget-priced, dual-channel function/arbitrary waveform generators — the SDG1005 and SDG1050 — which feature DDS (Direct Digital Synthesis) technology, providing stable, high precision, low distortion signals. In addition to sine waves, they also provide square waves with fast rising and falling edges, pulse and ramp waveforms, white noise, and arbitrary waveforms. The SDG Series has a clear, simple front panel, with a user-friendly layout and built-in help menu to simplify operation and adjustment. Also featured is built-in AM, FM, PM, PWM, and FSK modulation, without requiring an external modulating source. Both units can function as a 200 MHz frequency counter. USB device/host I/O allows for storage of waveforms and setups. SDG Series signal generators have two independent output channels with adjustable relative phase.

With a 125 MSa/s sample rate, 14-bit resolution, and 16 Kpts waveform record length, the SDG signal generators create precise waveforms. Modulation, sweep, and burst output modes are also provided. An extremely useful feature of the SDG Series is that they can interconnect with matching SDS1000 digital oscilloscopes for remote controlled testing and for directly reading and reconstructing stored scope waveforms.

Housed in a rugged 11" x 9" x 4" case with built-in prop stand, the SDG1005 and SDG1050 AWGs are available now for US $399 and $599, respectively.

For further information, please contact:

Saelig Company, Inc.
Tel: **888•7SAELIG**
Email: **info@saelig.com**
Website: **www.saelig.com**

Servo Mount Gears and Sprockets

ServoCity is now offering a way to drive gears and sprockets using their patented servo mount gears and sprockets that are able to attach directly to the output spline of a servo. These gears and sprockets are available in plastic or metal construction, and are available in both Hitec 24 tooth standard splines and Futaba 25 tooth standard splines. The metal gears and sprockets have broached splines to ensure a snug fit that will hold up in extreme applications. A large variety of gear sizes are available in both 32 and 48 pitch. Sprockets are

available for .250 (1/4") chain. These components are ideal for robotics applications, pan and tilt mechanisms, camera sliders, and animatronics applications.

Servo Protractor

Also available from ServoCity is the Servo Protractor which allows the user to precisely read the amount of rotation that a servo is able to travel. Using the servo protractor in conjunction with a Hitec servo programmer makes fine-tuning the rotation of digital servos easy. Simply take the horn/arm off of the servo, place the servo in the back of the servo protractor, and align it against the aluminum spacer. The spacer is in place so that each servo that's programmed is set exactly the same. The spacer can also be adjusted for servos of various sizes. This unit is recommended for Hitec servos with the standard 24 tooth Hitec spline.

For further information, please contact:

ServoCity	Website: **www.servocity.com**

ProtoSnap

SparkFun Electronics now has a new line of products designed to help the novice electronics enthusiast ease into the world of programming, prototyping, and design.

Each product in this new ProtoSnap line features various input and output boards that are linked together — complete with traces — to form a multi-use prototyping platform. This allows users to experiment with embedded electronics without the burden of soldering, wires, or other typical prototyping limitations.

There currently are three different products in the ProtoSnap line: the Pro Mini, the LilyPad Development Board, and the LilyPad E-sewing kit.

The Pro Mini combines an Arduino Pro Mini with a host of inputs and outputs to allow users to experiment with the Arduino language. When they have mastered programming the ProtoSnap Pro Mini, it can be broken apart so the individual components can be used separately. Both the ProtoSnap LilyPad Development Board and the ProtoSnap LilyPad E-sewing kits are designed to help users ease into e-textiles. They, too, can be broken apart into individual components and used in any number of different projects and applications.

For further information, please contact:

SparkFun Electronics	Website: **www.sparkfun.com**

Is your product innovative, less expensive, more functional, or just plain cool? If you have a new product that you would like us to run in our *New Products* section, please email a short description (300-500 words) and a photo of your product to:

newproducts@servomagazine.com

Then and NOW

MAN VS. MACHINE

by Tom Carroll

Man vs. machine ... which is better? Talk about a loaded question! I'm not talking about the popular series of arcade and smart phone games. I'm referring to on-going debate concerning the ability of a machine to emulate or surpass a human in many (or all) capabilities.

Fifteen years ago, IBM built a computer named Deep Blue (shown in **Figure 1**) and challenged chess Grandmaster Gary Kasparov to a series of chess games (**Figure 2**). Deep Blue won the first match but Kasparov returned to win three and draw two of the games, soundly defeating the computer. The next year, Deep Blue 'returned' and beat the Russian six games to zero. A decade and a half later in 2011, IBM was ready with another challenger and a different type of match — the TV quiz show, Jeopardy.

Jeopardy is quite different than a game of chess as it requires a very wide base of knowledge. As most of you know, in Jeopardy the answers are given to the contestants — most of the time in the form of puns or other word games. The new IBM computer was programmed to understand the complexity of word play in the human language and respond accordingly with a question. IBM's super-computer Watson (depicted as the center contestant in **Figure 3**) managed to best Ken Jennings and Brad Rutter — two of the winningest players of the TV game show. The appearance on the show was more than just a stunt; it displayed a specific capability of a computer to find the best answer for a certain question, or in this case, the best question for an answer. Did this experiment prove that a machine was better than a human?

FIGURE 1. IBM's Deep Blue.

Much the same as the Deep Blue competition, the actual computer was not what was on stage but was in a large server facility miles away. It drew 175 KW of power and filled the AC-cooled server room versus a measly 150 watts for each of the two human beings, with individual brains using only 10 watts each. Some might say that Watson did indeed beat the humans, but others have said that humans are still the better question answerers despite the loss. Maybe when Moore's Law progresses a bit (lot) more and compresses the required transistor mass down to a few kilograms drawing a few watts, then we'll have a true competitor for a human's intelligence.

New IBM Computer Core Chip Closer to Human Brain

Speaking of lower power and a better way to process information, this past August IBM made an announcement about a neurosynaptic chip or — as they call them — neural cores. Working with funding from DARPA and in conjunction with four US universities, the IBM chips work very much like the human brain. The neurons and synapses of the brain are interconnected with each other and just as our brains learn, these interconnections are always changing. This is different from standard computer chips where information is processed systematically on non-changing circuitry, one piece of data

FIGURE 2. Kasparov vs. Deep Blue.

at a time. These neural cores have the ability to be reconfigured when presented with a new task, just as our brains do. This fall at a custom IC conference, two papers are scheduled to be presented outlining how these chips have learned tasks such as playing the game of Pong, navigating a car on a clear pathway, and image recognition.

These neurosynaptic chips/neural cores are the earliest building blocks of what IBM hopes to develop into a more complete system — a cognitive computer. News from IBM's Cognitive Computing Group started making headlines in 2007 when researchers first simulated a mouse brain, then stepped up to a rat brain, then to a cat brain in 2008, and finally to a monkey brain. They soon realized that the power required by a supercomputer similar to Watson (to simulate human-scale reasoning) would incinerate itself if placed in the small scale of a human brain. The hundreds of kilowatts had to be reduced by a factor of ten thousand or more, down to our brain's power consumption of 10 watts or so. This new model that researchers were seeking would use these low power building blocks of the neural cores for the cognitive computer. The scientists involved freely admit that much physical and chemical knowledge has been learned about the human brain but exactly how it works to produce intelligence is the still-to-be-found "Holy Grail."

Artificial intelligence has long been one of the most interesting topics for robot experimenters, though some really prefer the term 'synthetic intelligence,' as we are well aware that our computer approach is not the way that our brains process information. However, these neural cores meld the central processing and random access memory (RAM) sections so close together on the silicon chips, that it is much the same as our brain's neuron and synapses. The resulting power requirements are minimal, and the re-configuring ability coupled with a cool-running chip base is the closest that we've come to a non-biological brain. The software used doesn't require the traditional programming of logic, rules, and step-wise sequences. It uses the interconnection of the neural cores so that the cognitive computer learns as do we. Learning is its programming. This has been a giant step for AI, but we humans are still on the top of the pyramid of intelligence. We can gaze down at Watson, the neural chip computers, and all of the other robots, but we must realize that intelligence equality will so be upon us.

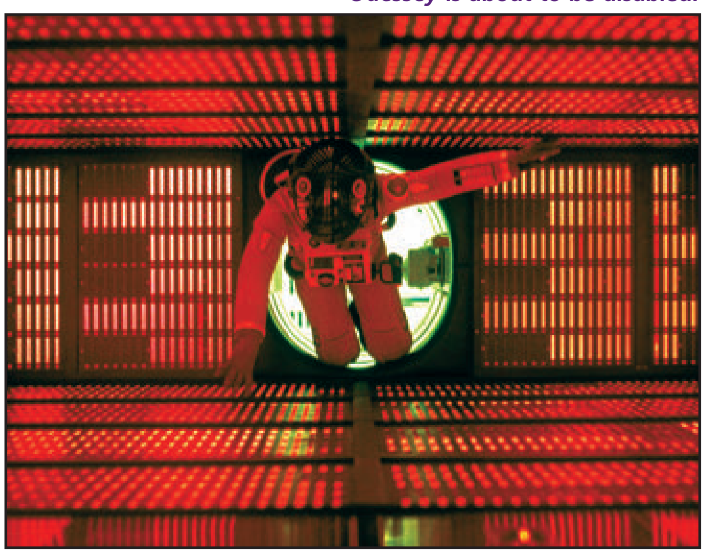
Early Ideas Expressing Machine Excellence

The earliest robot stories and movies always presented a bit of distrust of robots living among humans. Robots were always considered way too powerful to work along side humans, and many were banished to other planets as a safety precaution. These classic stories presented robots as potential enemies of Mankind, often about to take over Earth. Even later movies such as *iRobot* and the *Terminator* series showed robots as threats to humans. Isaac Asimov's stories of the '40s and '50s softened our perception of robots with the Three Laws of Robotics, but that didn't stop his 'iRobot' series of short stories from being twisted a bit in the Will Smith movie of the same name when US Robots' latest production run of robots seemed to go a bit crazy (to say the least). In the mid 1960's film, *2001 — A Space Odyssey*, surviving astronaut Dave Bowman is shown in **Figure 4** lobotomizing the spacecraft's computer, HAL 9000. It seems HAL decided that he could run the mission better than the five original crewmembers so he killed off four of them, and then tried to kill Bowman. Dave proved that man was more adept at running a spaceship than a machine, because he could disconnect the malfunctioning computer's memory.

It took businesses many decades to finally trust and accept computers. Robots followed the explosive growth of computers, and it's still taking some people a while to completely accept robots in industry, as well as in the home. Because of these misconceptions, many people continue to feel a bit of uneasiness with the prospect of a robot in their homes. However, I really doubt they think that their robot vacuum cleaner will attack while they sleep.

FIGURE 3. IBM's Watson on Jeopardy.

FIGURE 4. The HAL 9000 computer in *2001 — A Space Odyssey* is about to be disabled.



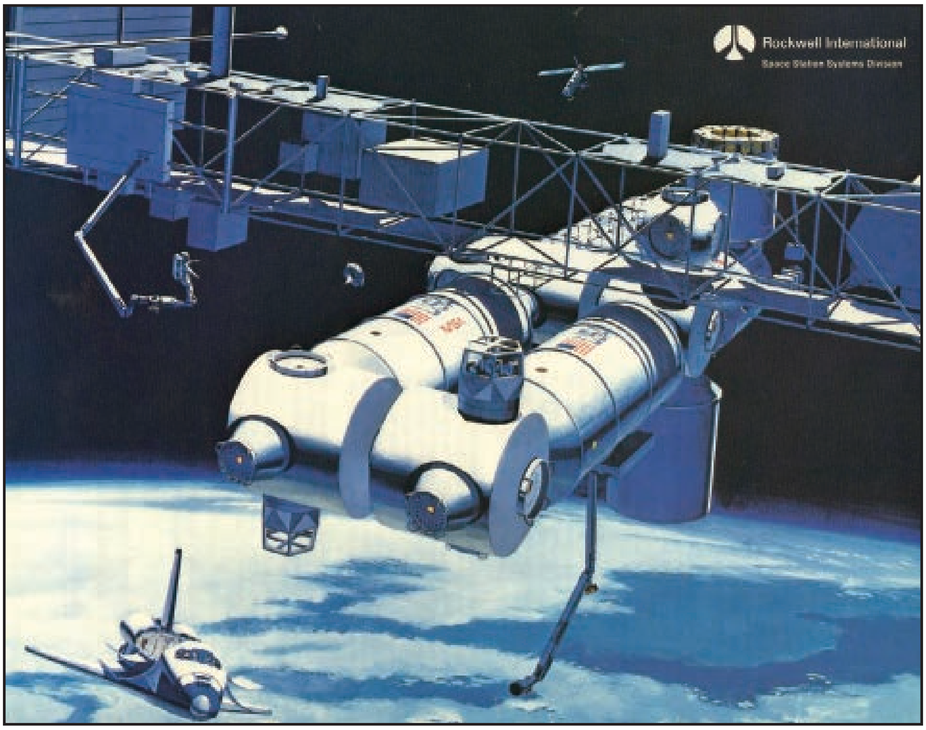


FIGURE 5. Space Station mobile remote manipulator system.

Robots Vie for Space Borne Tasks with Humans

Let's take a look at everyday tasks performed by both robots and humans. If you were employed by NASA and worked at Johnson Space Center in Houston, TX, you'd tend to lean towards man as more capable for most space borne tasks that NASA requires in their many varied missions, and that includes the Robonaut robot. You would stress that there is no AI program that even comes close to approximating the needed qualities required for a complex space-station construction procedure — even systems such as Watson. Space exploration is so unique and specialized that — if at all possible — placement of human astronauts at areas of interest is critical because humans have the natural ability to react to and solve unanticipated problems. Placing humans in space is costly, but so is building and placing applicable robotic systems. When I worked with Rockwell, we designed a Mobile Remote Manipulator System (MRMS) shown in **Figure 5** that was to be used on the NASA space station and would have cost a good portion of a billion dollars had it been constructed. It was designed to crawl about the exterior of the space station in a ladder-climbing fashion and bring supplies to various locations, or to assist in docking of the shuttle and payload maneuvering. My intention in the design was not to replace astronauts, but to assist them in very difficult tasks outside the station's pressurized sections. Yes, these tele-operated robots can certainly be of great value to astronauts, but you would still want man to be in the loop.

Head out west to Pasadena, CA and NASA's Jet Propulsion Labs and you'd hear just how capable machines are in carrying out tasks in deep space and planetary surfaces. JPL personnel would insist that bringing man to the various million or billion mile distant sites of interest would be prohibitively expensive. "People require food, fluids, oxygen, a narrow temperature range, and an even more narrow atmospheric pressure range" they would say. "Besides, people and their bodily needs weigh too much to send into space." These are very valid points and the very successful Martian rovers have operated 28 times their expected lifetimes.

Robots Excel in Certain Tasks on Earth

Getting back to Earth, go to an automobile assembly plant and you'll see humans working alongside hulking robots, building cars with great efficiency. Sparks fly from the large spot-welding robots and noxious paint is sprayed from others. Robots deftly extract hot metal castings from an annealing furnace and place them on a pallet with the human controller watching safely from the side. Robots do not necessarily take jobs away from workers. Robots take over the dangerous, unpleasant, and monotonous tasks that people don't want, freeing them to be robot operators, repair persons, and programmers.

People Also Excel in Certain Tasks on Earth

Go across town to an aerospace plant where multi-million dollar spacecraft are produced and you'll see hundreds of humans in clean-room attire carefully assembling precision components all by hand. Precisely-machined 'fixtures' delicately hold and maneuver the space system's structures in production, allowing the workers access to the different sections. Most of the maneuvering is accomplished by human power. Engineering and manufacturing personnel quickly learned that no robot existed that would allow them to assemble their extremely costly products, and designing and building a robot for a few specialized products that the company would manufacture would not be a cost-effective measure. **Figure 6** shows technicians at JPL assembling the robot rover, Curiosity in a clean room. Robotic assistance for these types of operations and product assembly would be way too expensive to implement.

This conundrum has befuddled mankind for many decades, even centuries. Long before robots were ever present in our factories, mechanization was a solution to assist humans in many tasks. Blacksmiths gladly dispensed with hand and foot powered bellows that provided blasts

of air to keep their fires hot, and welcomed blowers powered by steam engines and (later) electric motors. Hand powered looms in textile mills were eventually powered by wheels turned by flowing streams of water. In fact, it was this last example that brought about the industrial age here in the United States. Many towns bloomed in the New England states due to the proximity of reliable streams nearby.

Modern Day Robots Assist Humans in Dangerous Situations

As I've touched upon in this article, robots work best when they assist humans in day to day mundane tasks. Of course, the tasks can be as difficult as working with a police department in a dangerous hostage or bomb situation. Quite often the robot does not have to be superior to the human operator but rather just expendable. Most robots — size and weight being equal — are far less powerful than a human being. Where humans are 'inferior' to an equivalent robot is in their ability to take a bullet wound or exhibit long physical endurance. A 'dead' robot is far easier to deal with than a dead or injured police officer.

Figure 7 shows a Remotec Andros F6-A 485 pound law enforcement robot on display at the 2005 AUVSI exhibition. This robot is in the mid-range of law enforcement robots produced by Remotec, with models ranging from an HD-1 at 200 pounds to the largest — the Wolverine — at 810 pounds. The F6-A has probably garnered the most news and TV show coverage of any law enforcement robot.

With the current wars taking many American military personnel lives, the use of robots is becoming more prevalent these days. A new statistic reveals that for every 50 soldiers on the battlefield, there is a robot out there with them — both on the ground and in the air. A battlefield is always a dangerous place and the military robot quite often carries one or more live weapons to deal with a deadly enemy. One of the robots used recently is the Talon robot by Foster Miller (see **Figure 8**). These war machines can carry an M16 rifle, a 5.56 mm SAW M249, a 7.62 mm M240 machine gun, a .50 cal M82 Barrett rifle, a six barreled 40 mm grenade launcher, or quad 66 mm M202A1 FLASH incendiary weapon. Traveling at 5.2 MPH, they can keep up with the soldiers and go forward under fire as the men take cover and maneuver the robot with a simple hand controller. At $230K each, they are not cheap but are worth the cost to save a human life. For every bullet that the robot takes, there's one less soldier brought home to a grieving family.

FIGURE 6. Mars Rover, Curiosity, in JPL clean room.

Surgical Robots

This last class of robots deals a bit closer to home than I'd like. This past spring, I was diagnosed with prostate cancer — not something I wanted to hear. After a bit of soul-searching and a vast amount of research on the Internet, I had a prostatectomy via robot-assisted surgery this September. My urologist trained extensively on the daVinci Surgical System back east and has done numerous surgeries with it. **Figure 9** is from a daVinci advertisement and shows the surgeon's console in the foreground and the patient cart in the back with the three 'endoWrist' arms and a 3D camera. Looking at it reminds me of the so-called robot plastic surgeon in the film *Logan's Run* that went wild in one scene, cutting up a patient. Surgeons typically use these systems for prostatectomies, cardiac valve repair, and gynecologic procedures — mainly because of the minimally invasive results compared with radical surgery. (I have been asked by *SERVO* to do a feature article on my experiences and results of the surgery, so stay tuned.)

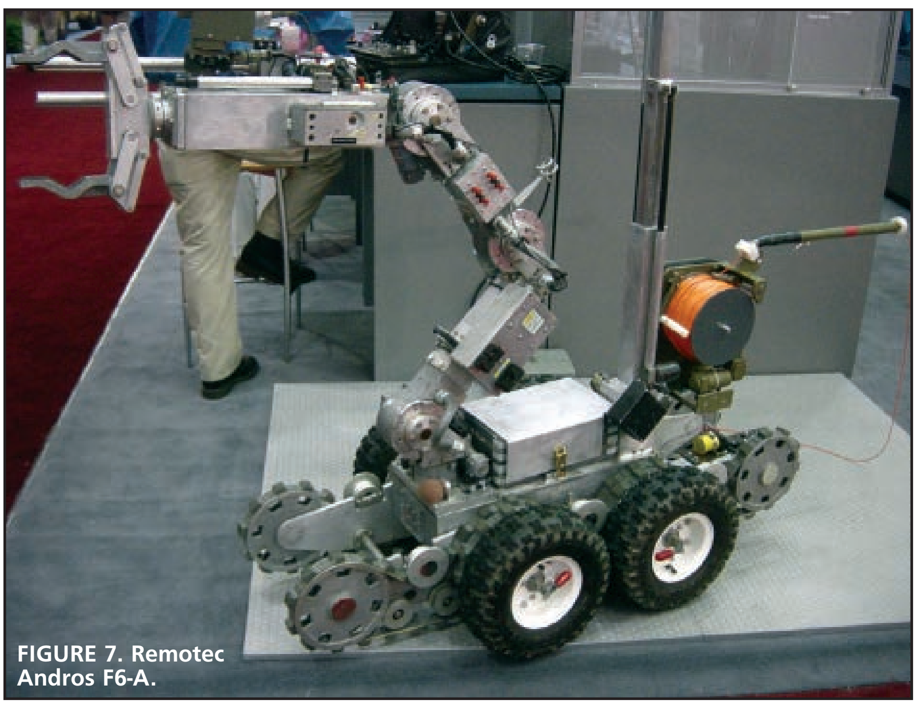


FIGURE 7. Remotec Andros F6-A.

FIGURE 8. Talon robot.

Final Thoughts

I've talked about some widely diverse types of robots. All of these types have superior capabilities but also inferior capabilities as compared to humans. The 'superior' IBM computers mentioned at the beginning really don't have even a small percentage of the capability of a human brain, despite trouncing their human competitors in games. The rest of the robots that I covered were really teleoperated with the true intelligence supplied by a human.

Does this mean that we will always be superior to robots? No, not really. IBM and other companies, as well as universities, are developing robots that are rapidly developing very human-like capabilities. The day will come when we might have to admit that human-produced robots have finally exceeded their maker's capabilities. Not in my lifetime, though. I welcome your feedback, as always. **SV**

FIGURE 9. daVinci surgical robot.

Tom Carroll can be reached at TWCarroll@aol.com.

ROBO-LINKS

Advertiser Index


NO MATTER WHAT THE IDEA
YOUR PCB PROTOTYPES SHOULD BE THE EASY PART
QUOTE & ORDER PCBS ONLINE AT WWW.SUNSTONE.COM OR CALL 1-800-228-8198